

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____ 2019 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ**

**на тему
АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ТРЕНАЖЕРА «МЕТОД
РЕЗОЛЮЦІЙ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ
«МАТЕМАТИЧНА ЛОГІКА»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Сосновський Дмитро Романович

_____ «___» _____ 2019р.
(підпис)

Науковий керівник к.ф.-м.н., доц. Черненко Оксана Олексіївна

_____ «___» _____ 2019р.
(підпис)

ПОЛТАВА 2019р.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	4
1. ПОСТАНОВКА ЗАДАЧІ.....	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	8
2.1. Огляд тренажерів з математичних дисциплін	8
2.2. Позитивні і негативні аспекти оглянутих тренажерів	14
2.3. Актуальність теми роботи.....	14
3. ТЕОРЕТИЧНА ЧАСТИНА	16
3.1. Метод резолюцій.....	16
3.2. Приклади застосування методу резолюцій	19
3.3. Алгоритмізація за темою роботи.....	23
3.4. Розробка блок-схеми.....	36
4. ПРАКТИЧНА ЧАСТИНА	48
4.1. Обґрунтування вибору програмних засобів.....	48
4.2. Опис процесу програмної реалізації	50
4.3. Опис програми.....	55
4.4. Інструкція по використанню тренажера.....	60
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТОК А. КОД ПРОГРАМИ	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
Java	об'єктно-орієнтована мова програмування
апплет	прикладна програма, написана на мові програмування Java у формі байт-коду
NetBeans IDE	середовище розробки
правило резолюцій	правило висновування, що сходить до методу доказу теорем через пошук протиріч; використовується в логіці висловлювань і логіці предикатів першого порядку
диз'юнкти	диз'юнкцій пропозиційних змінних чи їх заперечень

ВСТУП

Для того, щоб студент зміг успішно виконувати свої функції у професійній діяльності, йому необхідно під час навчання набути певних практичних навичок та теоретичних знань. Однією із важливих умов підвищення розвитку теорії і практики є створення системи завдань, їх послідовність та різноманітність.

Під комп'ютерним тренажером розуміється навчально-тренувальний пристрій, який імітує обставини, дії, створює ситуацію, наближену до реальної. У більш вузькому значенні це комп'ютерна навчальна програма для розвитку у студентів умінь та навичок певної діяльності, а також розвитку пов'язаних з нею здібностей.

Метою роботи є алгоритмізація елементів тренажера з теми «Метод резолюцій» дистанційного навчального курсу «Математична логіка» та закріплення знань з використання алгебри предикатів.

Об'єктом розробки є процес дистанційного навчання математичним дисциплінам.

Предмет розробки – програмний продукт, що реалізує тренажер для закріплення знань із використання методу резолюцій.

Головне завдання – розробити алгоритм роботи тренажера, по якому потрібно розробити навчальний тренажер з теми «Метод резолюцій», як складову дистанційного курсу «Математична логіка».

Методи розробки – методика застосування методу резолюцій в численні висловлень і численні предикатів.

При реалізації програми використано мову програмування Java і середовище розробки NetBeans IDE. Тренажер розроблено з використанням Java-applet.

Тренажер готовий до використання в дистанційному курсі «Математична логіка». Щоб іноземні студенти теж могли використовувати програму надається можливість вибору мови.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі для реалізації тренажера. У другому розділі описано огляд тренажерів з математичних дисциплін, їх позитивні та негативні аспекти, актуальність теми роботи. У третьому розділі представлено теоретичний матеріал з методу резолюцій, наведено приклади застосування та описано алгоритм роботи тренажеру, представлено блок-схему алгоритму. У четвертому розділі – описано обґрунтування вибору програмних засобів, процес програмної реалізації, саму програму і інструкцію по використанню тренажера.

Обсяг пояснювальної записки: 80 стор., в т.ч. основна частина - 59 стор., джерела - 14 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Оскільки потрібно розробити навчальний тренажер з теми «Метод резолюцій», як складову дистанційного курсу «Математична логіка», то слід виконати наступні завдання роботи:

- розглянути тренажери схожої тематики з математичних дисциплін;
- визначити всі аспекти оглянутих тренажерів та вказати актуальність теми роботи;
- розглянути теоретичні відомості про метод резолюцій;
- розглянути специфіку застосування методу резолюцій в численні висловлень і численні предикатів;
- розробити алгоритм тренажера, що дозволить закріпити знання з теми «Метод резолюцій»;
- обґрунтувати вибір програмного забезпечення;
- описати процес програмної реалізації;
- описати роботу тренажера.

Щоб допомогти студенту у проходженні тренінгу потрібно розробити можливість звернутися до теоретичного матеріалу з теми. В алгоритмі слід покроково описати всі дії при розв'язанні прикладів, де необхідно застосувати метод резолюцій в численні висловлень та метод резолюцій в численні предикатів.

Слід реалізувати перехід від української версії тренажера до англійської для можливості його використання іноземними студентами як додаткового матеріалу при опрацюванні теми «Метод резолюцій». Весь вміст програми повинен виводитися на двох мовах.

Розроблений тренажер повинен мати зрозумілий для користувача інтерфейс.

Пропонується, щоб він містив:

- стартову сторінку;
- сторінку з теоретичним матеріалом або окремий файл;
- сторінку з умовою прикладу, завданням, варіантами відповідей;
- кінцеву сторінку.

Сторінка з теоретичним матеріалом має бути доступною користувачеві постійно, щоб він міг в будь-який момент до неї звернутися.

Якщо користувач невірно відповість на завдання, то повинно виводитися повідомлення про помилку і слід вказати вірну відповідь.

При завершенні проходження тренажера необхідно реалізувати можливість розпочати проходження спочатку або завершити його.

Результати роботи слід опублікувати в матеріалах науково-практичного семінару і збірнику наукових статей магістрів.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд тренажерів з математичних дисциплін

В тренажері з теми «Переставні многогранники» дистанційного курсу «Елементи комбінаторної оптимізації» на стартовій сторінці виводиться назва тренажера, вказується розробник і керівник (рис. 2.1) [2-4].

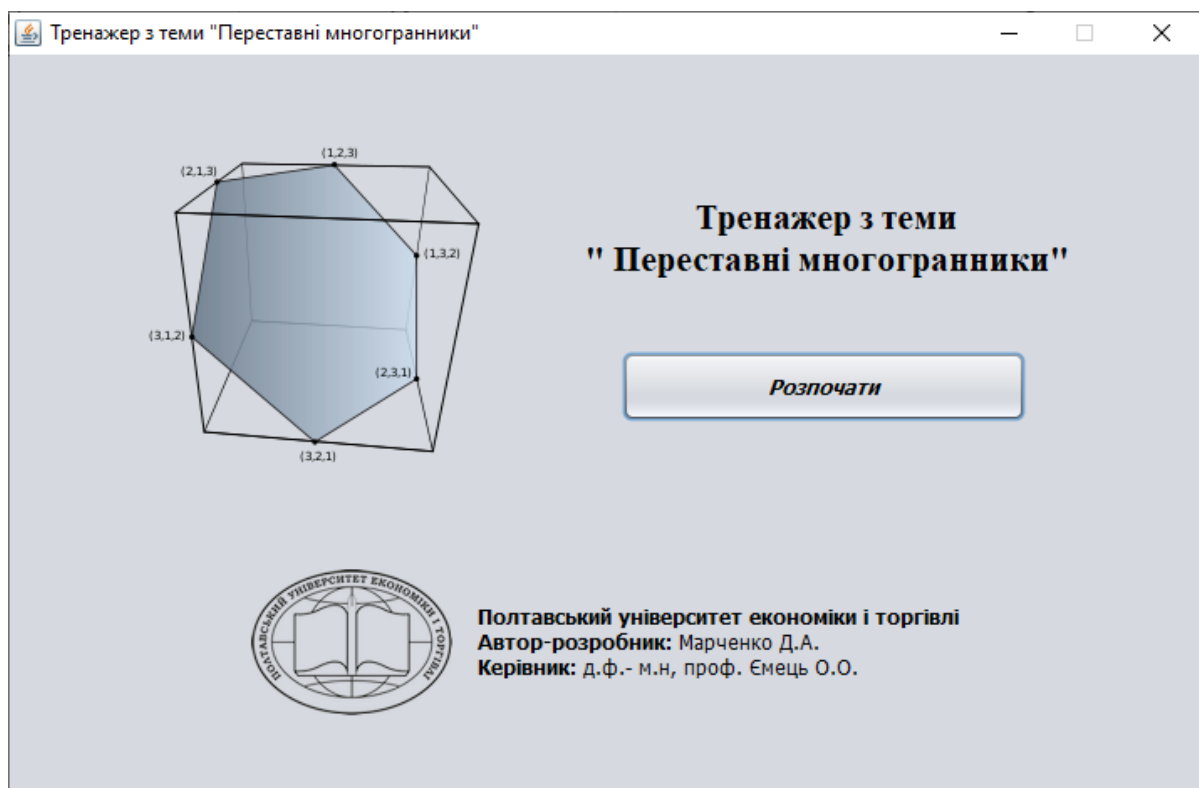


Рисунок 2.1 – Стартова сторінка тренажера з теми «Переставні многогранники»

Наступним кроком виводиться постановка задачі і надається вибір множини і параметрів (рис. 2.2). Всі кроки представлені завданнями з вибором однієї відповіді (рис. 2.3).

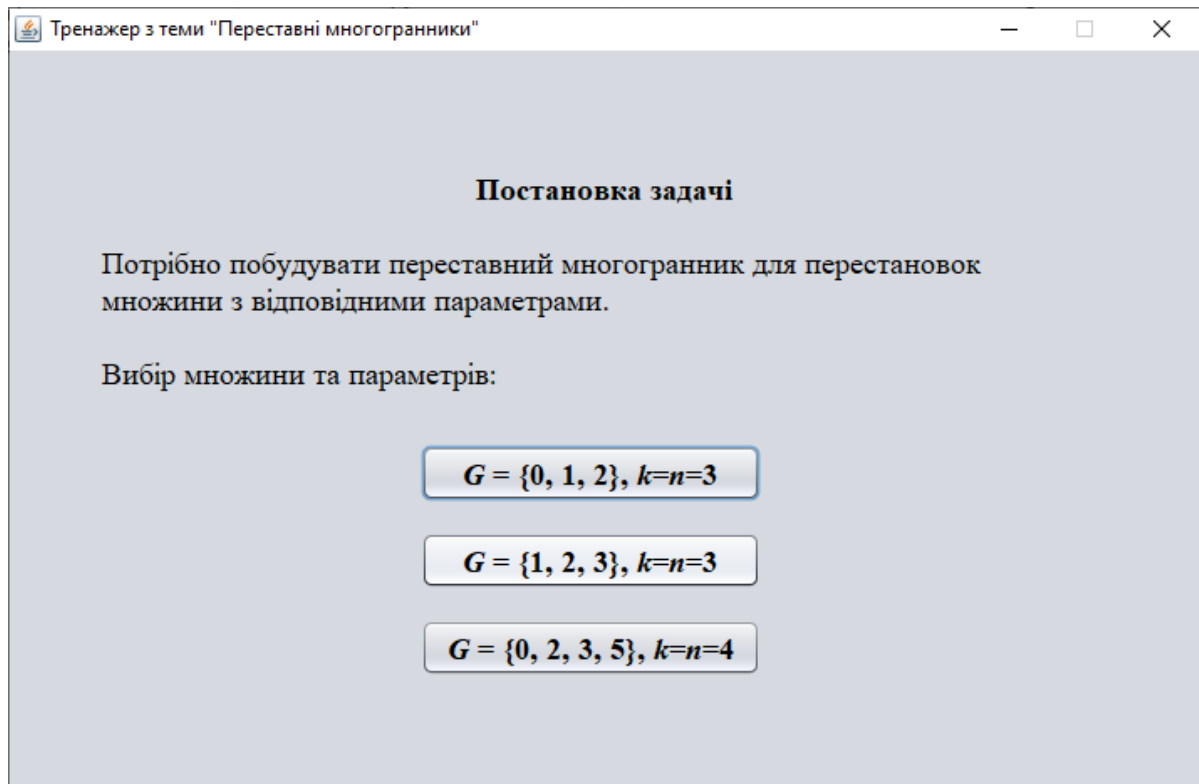


Рисунок 2.2 – Постановка задачі тренажера з теми «Переставні многогранники»

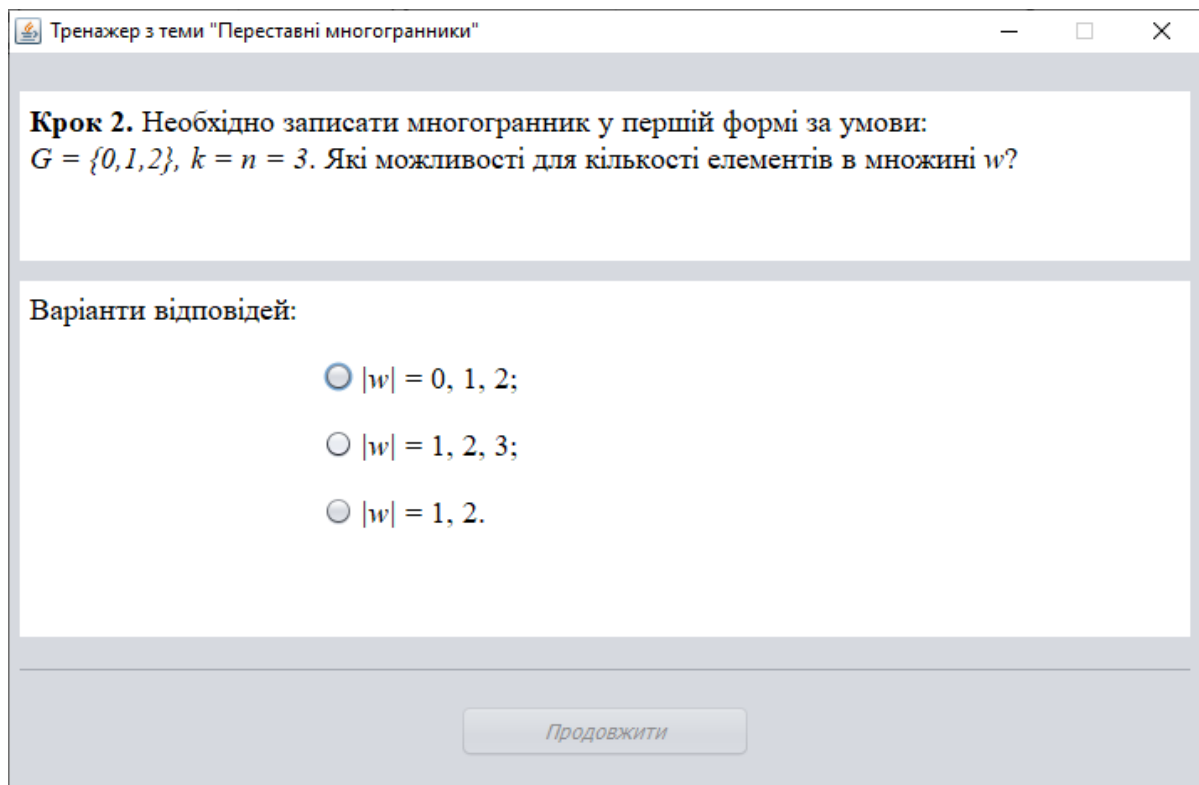


Рисунок 2.3 – Завдання в тренажері з теми «Переставні многогранники»

В кінці тренажера виводиться кількість зроблених помилок, перелік кроків, надається можливість вибрати іншу множину та пройти тренажер знову (рис. 2.4).

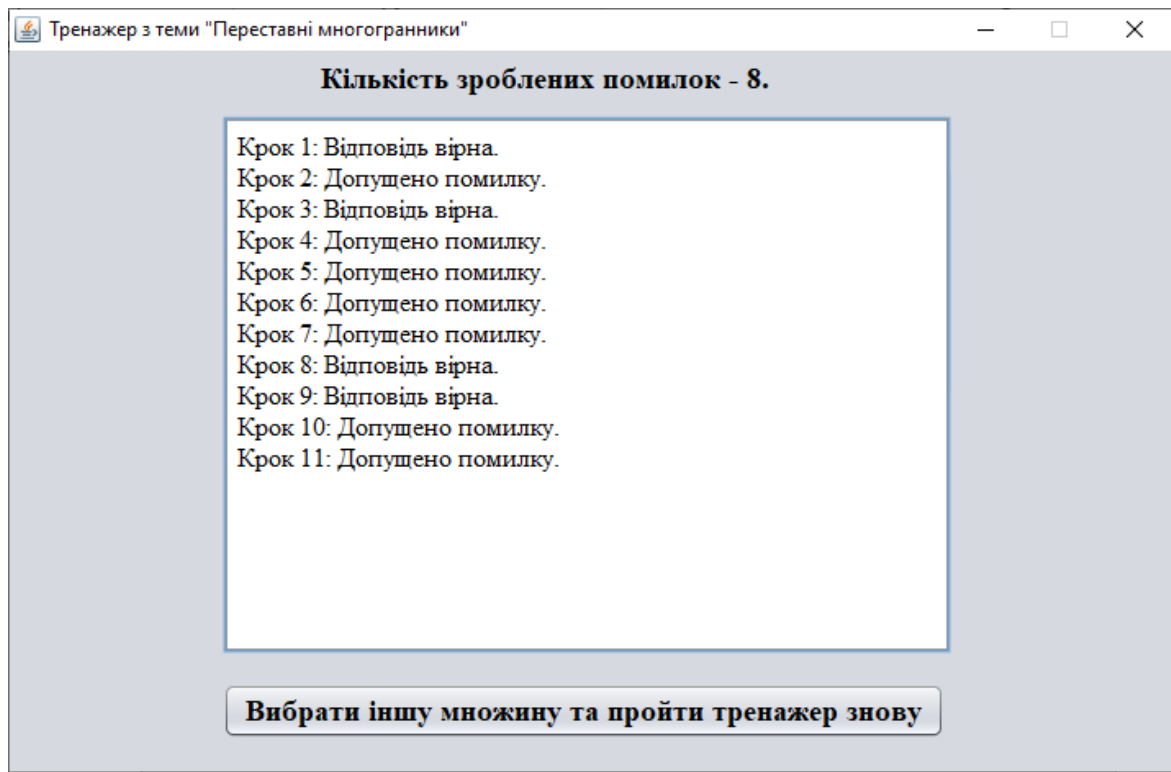


Рисунок 2.4 – Кінець тренажера з теми «Переставні многогранники»

Наступний тренажер з теми «Гرادієнтний метод» дистанційного курсу «Методи оптимізації та дослідження операцій» (рис. 2.5) [5-7].

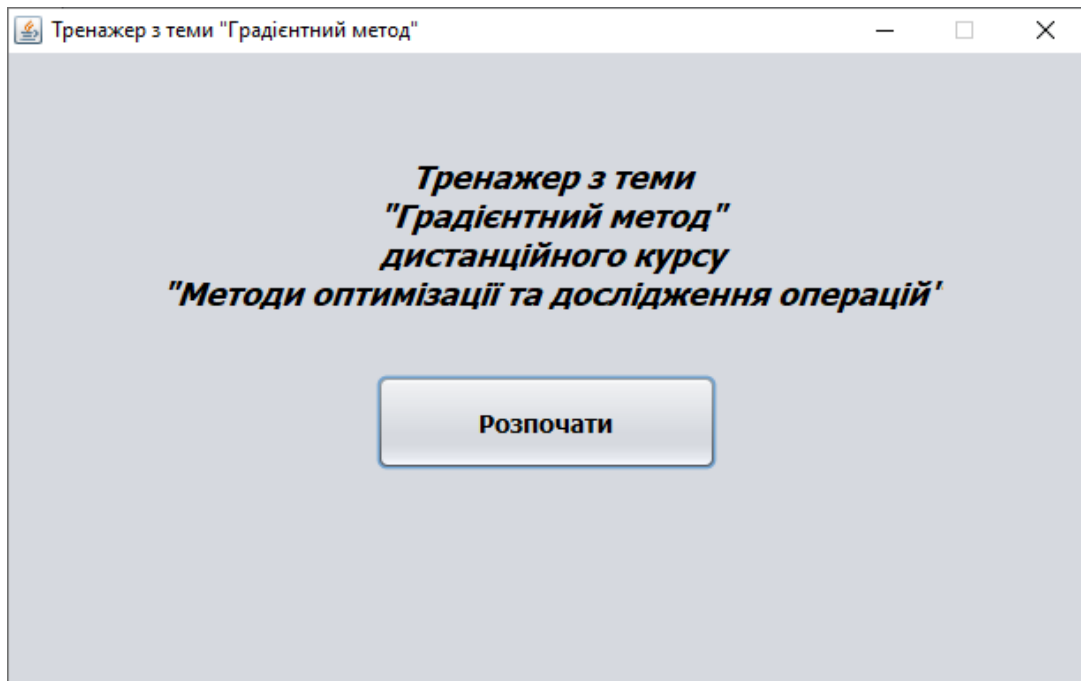
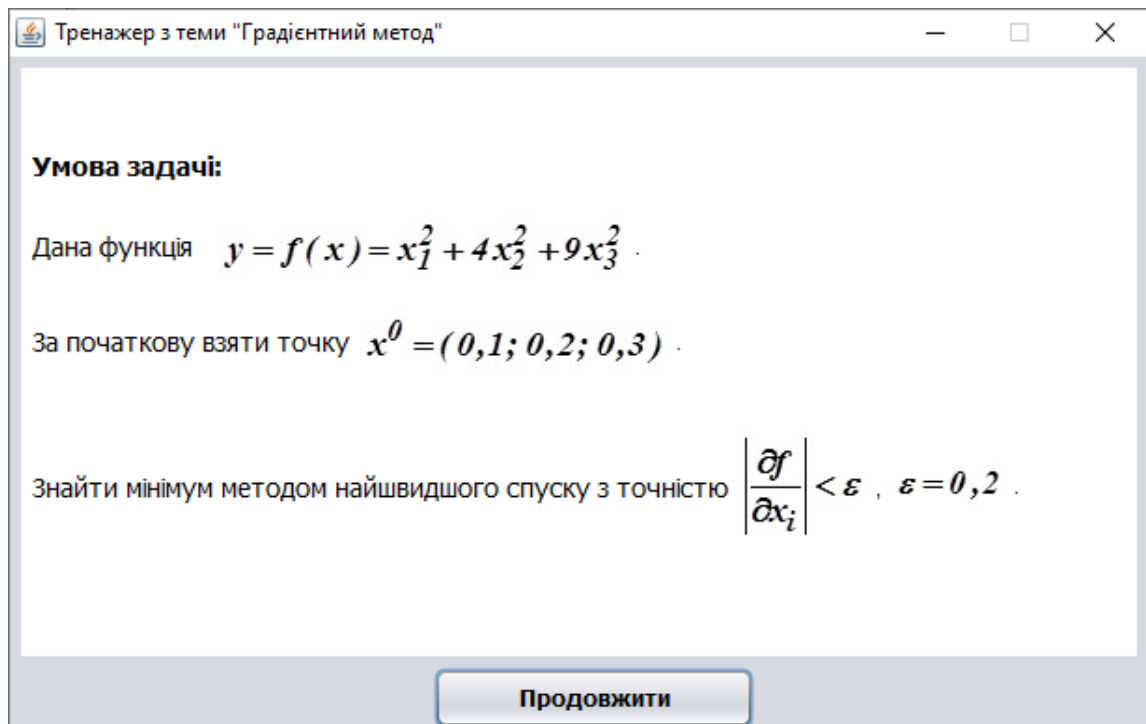


Рисунок 2.5 – Стартова сторінка тренажера з теми «Градiєнтний метод»

Спочатку виводиться умова задачі, а потім завдання (рис. 2.6). Завдання пропонують як вибір однієї з наведених відповідей (рис. 2.7), так і заповнення комірок (рис. 2.8).



Умова задачі:

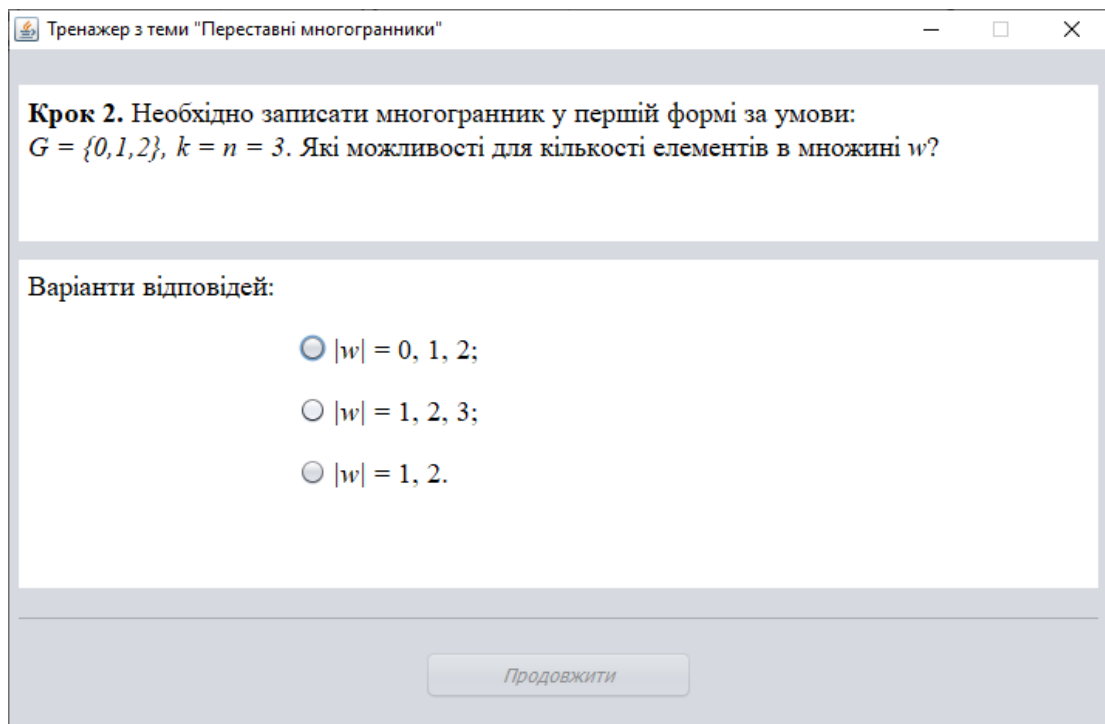
Дана функція $y = f(x) = x_1^2 + 4x_2^2 + 9x_3^2$.

За початкову взяти точку $x^0 = (0,1; 0,2; 0,3)$.

Знайти мінімум методом найшвидшого спуску з точністю $\left| \frac{\partial f}{\partial x_i} \right| < \varepsilon$, $\varepsilon = 0,2$.

Продовжити

Рисунок 2.6 – Умова задачі в тренажері з теми «Переставні многогранники»



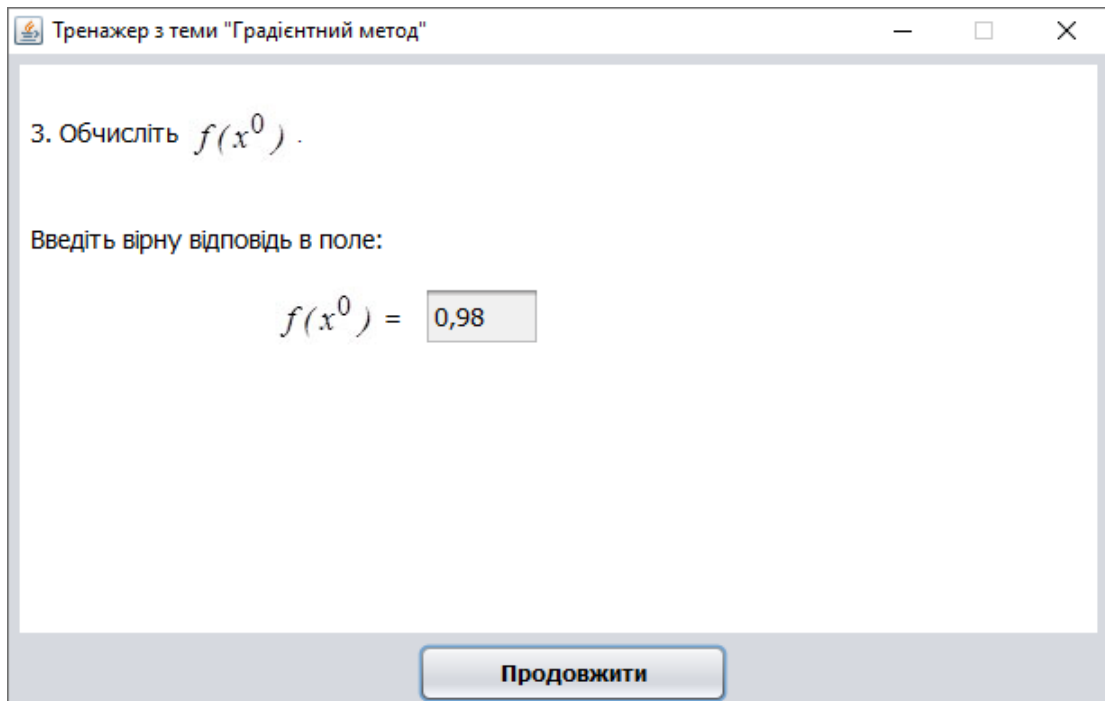
Крок 2. Необхідно записати многогранник у першій формі за умови:
 $G = \{0,1,2\}$, $k = n = 3$. Які можливості для кількості елементів в множині w ?

Варіанти відповідей:

- ☒ $|w| = 0, 1, 2;$
- ☐ $|w| = 1, 2, 3;$
- ☐ $|w| = 1, 2.$

Продовжити

Рисунок 2.7 – Завдання з вибором відповіді в тренажері з теми «Переставні многогранники»



Тренажер з теми "Градiєнтний метод"

3. Обчисліть $f(x^0)$.

Введіть вірну відповідь в поле:

$f(x^0) =$

Продовжити

Рисунок 2.8 – Завдання з заповненням комірки в тренажері з теми
«Переставні многогранники»

Як висновок відображається повідомлення про завершення роботи тренажера (рис. 2.9).

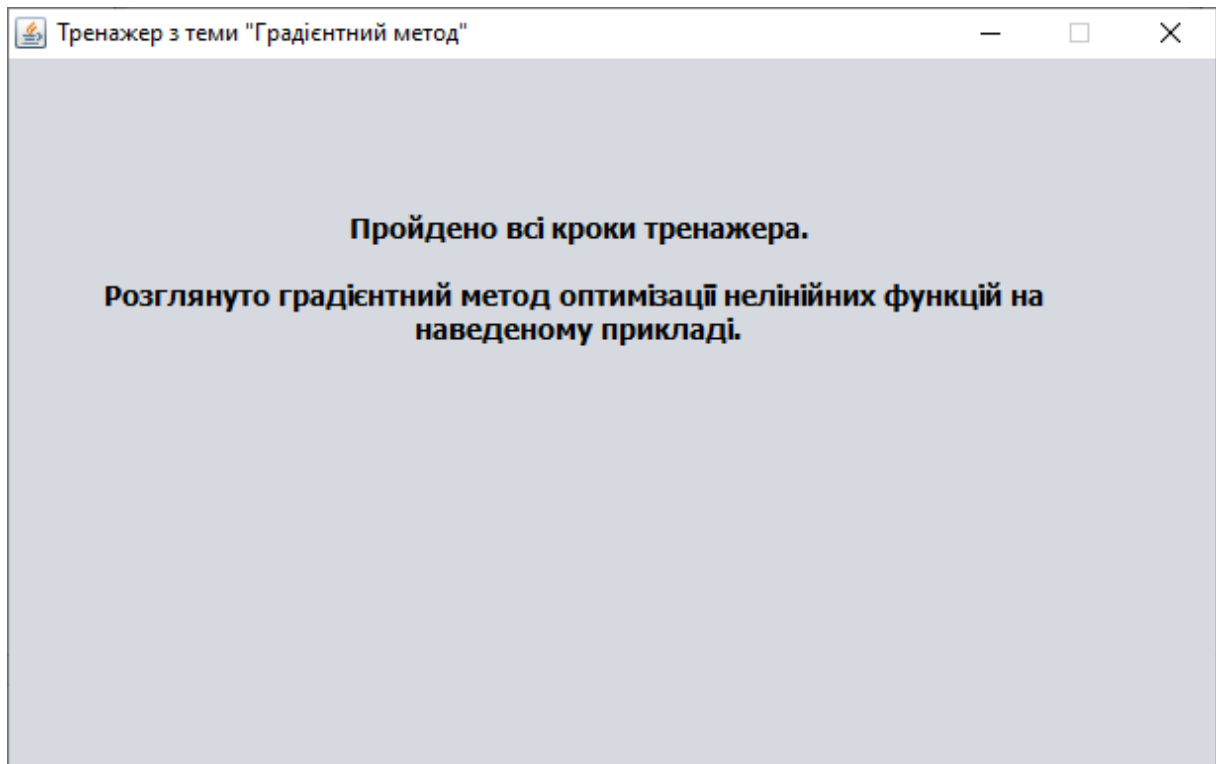


Рисунок 2.9 – Повідомлення про завершення роботи тренажера з теми
«Переставні многогранники»

Ще один тренажер з дисципліни «Методи оптимізації та дослідження операцій» ознайомлює користувача з методом гілок та меж в задачі про найкоротший шлях (рис. 2.10) [5].

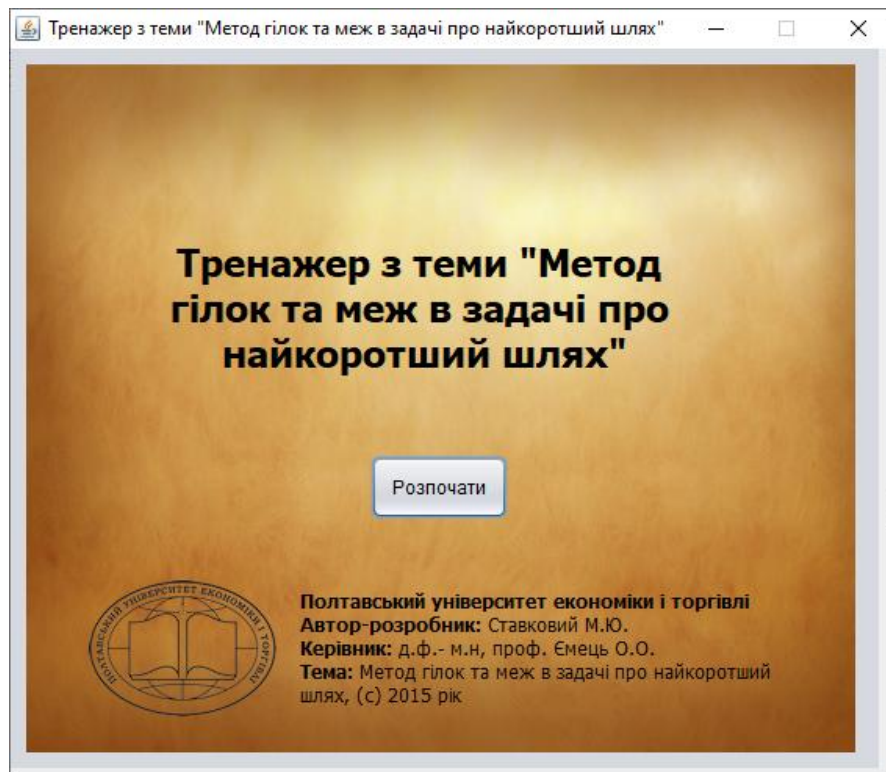


Рисунок 2.10 – Стартова сторінка тренажера з теми «Метод гілок та меж в задачі про найкоротший шлях»

Також пропонується вибір одного з прикладів (рис. 2.11). Кроки тренажера представлені завданнями з вибором однієї відповіді.

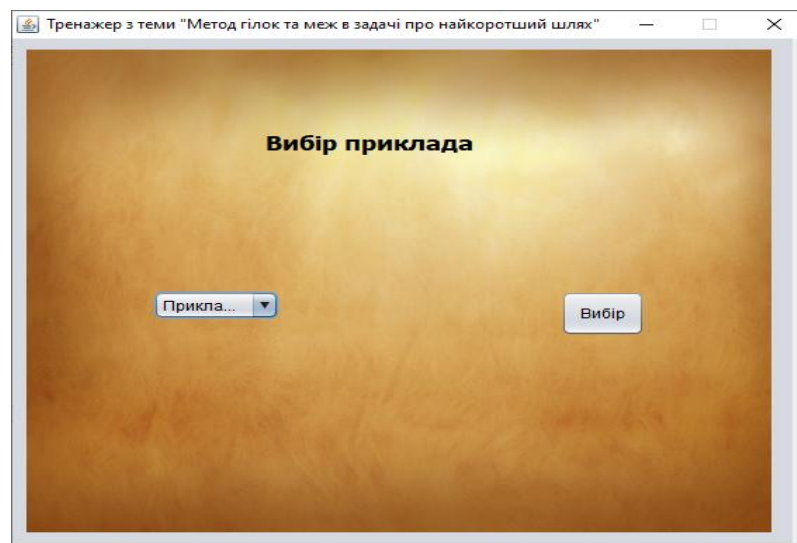


Рисунок 2.11 – Вибір прикладу в тренажері з теми «Метод гілок та меж в задачі про найкоротший шлях»

2.2. Позитивні і негативні аспекти оглянутих тренажерів

Оглянуті тренажери мають як позитивні, так і негативні аспекти. До позитивних можна віднести:

- покрокове розв'язання прикладу, що дозволить краще закріпити знання з теми;
- наявність декількох прикладів для різноманіття;
- відсутні лишні елементи, занадто пустий простір;
- відображення пройдених кроків і де було допущено помилки;
- можливість розпочати проходження тренажера спочатку;
- виведення помилки у разі неправильної відповіді.

До негативних можна віднести:

- відсутність достатньої інформації про тренажер;
- автоматична зміна на правильну відповідь після виведення повідомлення про помилку;
- недостатня інформативність повідомлення про помилку;
- відсутність можливості переходу на початок програми після повідомлення про завершення, потрібно перезапустити тренажер.

2.3. Актуальність теми роботи

На сучасному етапі розвитку інтелектуальних систем, які можуть використовуватись для вирішення широкого спектру задач, гостро стоїть питання оптимізації та прискорення машин логічного виведення. Сучасні програмно-апаратні засоби дозволяють розпаралелювати вирішення задач, тому розробка спеціальних, паралельних алгоритмів логічного виведення є актуальною проблемою.

Числення висловлень є важливою і невід'ємною складовою частиною всіх числень математичної логіки. Однак воно є занадто бідною для опису та аналізу найпростіших логічних міркувань науки і практики.

Однією з причин цього є те, що у численні висловлень будь-яке просте висловлення розглядається як вихідний об'єкт дослідження, неподільне ціле, позбавлене частин і внутрішньої структури, яке має лише одну властивість -бути або істинним, або хибним.

Зауважимо, що в логіці є два формулювання одержання правильних умовиводів. Перше постає у вигляді правил виведення, а друге – у вигляді логічних законів.

Логічні правила — це своєрідні директивні вказівки, які базуються на логічних законах і дають змогу визнавати правильними висловлення, що утворені в результаті виведення з істинних посилок.

Законами логіки висловлень і предикатів називаються схеми побудови істинних складних висловлень. Інакше кажучи, закони логіки висловлень і предикатів — це такі вирази, яким за будь-яких підстановок значень замість змінних завжди відповідає істинне висловлення.

Найбільш відомим методом логічного виведення є метод резолюцій, який полягає в почерговому утворенні резольвент. Недолік принципу резолюції полягає у формуванні на кожному кроці виводу безлічі резольвентів – нових диз'юнктів, більшість з яких виявляються зайвими. У зв'язку з цим розроблені різні модифікації принципу резолюції, що використовують більш ефективні стратегії пошуку і різного роду обмеження на вид вихідних диз'юнктів.

Логіка предикатів першого порядку для тотожно-істинної (несуперечливої) множини диз'юнктів, заснована на принципі резолюції, може працювати нескінченно довго.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Метод резолюцій

Правило резолюцій — це правило висновування, що сходить до методу доказу теорем через пошук протиріч; використовується в логіці висловлювань і логіці предикатів першого порядку. Правило резолюцій, що застосовується послідовно для списку резольвент, дозволяє відповісти на питання, чи існує у вхідній множині логічних виразів протиріччя.

Правило резолюцій першочергово застосовується до диз'юнктивів — диз'юнкцій пропозиційних змінних чи їх заперечень. Правило застосовується до двох диз'юнктивів у випадку, коли вони містять пропозиційні змінні, одна з яких є запереченням другої.

Правило резолюцій можна визначити таким чином:

$$\frac{\{C, p\} \{C', \neg p\}}{\{C, C''\}}.$$

Метод резолюцій запропонований у 1930 р. в докторській дисертації Жака Ербрана для доведення теорем у формальних системах першого порядку. Метод резолюцій спирається на обчислення резольвент. Існує теорема, яка стверджує, що питання про доказовість довільної формули в численні предикатів зводиться до питання про вивідність порожнього списку в обчисленні резольвент. Тому доведення того, що список формул в обчисленні резольвент порожній, еквівалентне доведенню хибності формули в численні предикатів.

Рішення в логічній моделі на основі методу резолюцій Дано твердження:

«Сократ — людина»;

«Людина — це жива істота»;

«Всі живі істоти смертні».

Потрібно методом резолюцій довести твердження «Сократ смертний».

Рішення:

Крок 1. Перетворимо висловлювання на диз'юнктивну форму;

Крок 2. Запишемо заперечення цільового виразу (необхідного виводу), тобто «Сократ безсмертний»;

Крок 3. Скласти кон'юнкцію всіх диз'юнктів, включивши в неї заперечення цільового виразу;

Крок 4. У циклі проведемо операцію пошуку резольвентів над кожною парою диз'юнктів;

Отримання порожнього диз'юнкта означає, що висловлювання «Сократ безсмертний» — хибне, значить, істинне висловлювання «Сократ смертний».

Метод резолюції відноситься до напівконструктивного методу, він легко піддається алгоритмізації. Суть його полягає в тому, що два посилкових диз'юнкта з протилежними термами завжди можна склеїти в один заключний диз'юнкт, в якому відсутні протилежні терми:

$$A \vee B, B \vee C = A \vee C$$

де A, C - довільні терми, або цілі диз'юнкти з довільним набором термів, включаючи нуль, а також - довільні терми. При послідовному застосуванні принципу резолюцій зменшується число букв, деякі повністю знищуються, а вихідна клауза будується у формі кон'юнктивного протиріччя:

$$P_1, P_2, \dots, P_n = 0$$

Доказ теорем зводиться до доказу того, що деяка формула G (гіпотеза теореми) є логічним наслідком множини формул F_1, \dots, F_k (припущень). Тобто сама теорема може бути сформульована таким чином: «якщо F_1, \dots, F_k істинні, то істинна і G ».

Для доказу того, що формула G є логічним наслідком множини формул F_1, \dots, F_k , метод резолюцій застосовується наступним чином.

Спочатку складається множина формул $fF_1, \dots, F_k, \neg Gg$. Потім кожна з цих формул приводиться до КНФ(кон'юнкція диз'юнктивів) і в отриманих формулах закреслюються знаки кон'юнкції. Виходить множина диз'юнктивів S . І, нарешті, шукається висновок порожнього диз'юнкта з S . Якщо порожній диз'юнкт виводимо з S , то формула G є логічним наслідком формул F_1, \dots, F_k . Якщо з S не можна вивести $\#$, то G не є логічним наслідком формул F_1, \dots, F_k . Такий метод доведення теорем називається методом резолюцій.

Розглянемо приклад докази методом резолюцій. Нехай у нас є наступні твердження:

:«Яблуко червоне і ароматне.»

:«Якщо яблуко червоне, то яблуко смачне.»

Доведемо твердження «яблуко смачне». Введемо множину формул, що описують прості висловлювання, відповідні вищенаведеним твердженням. нехай

: X_1 – «Яблуко червоне»,

: X_2 – «Яблуко ароматне»,

: X_3 – «Яблуко смачне».

Тоді самі твердження можна записати у вигляді складних формул:

: $X_1 \& X_2$ – «Яблуко червоне і ароматне».

: $X_1 \rightarrow X_3$ – «Якщо яблуко червоне, то яблуко смачне».

Тоді твердження, яке треба довести, виражається формулою X_3 .

Отже, доведемо, що X_3 є логічним наслідком $(X_1 \& X_2)$ і $(X_1 \rightarrow X_3)$.

Спочатку складаємо множину формул з запереченням доказуваного висловлювання; отримуємо:

$$f(X_1 \& X_2), (X_1 \rightarrow X_3), \neg X_3g.$$

Тепер приводимо всі формули до кон'юнктивної нормальної форми і закреслюємо кон'юнкції. Отримуємо наступну множину диз'юнктивів:

$$fX_1, X_2, (\neg X_1 \vee X_3), \neg X_3g.$$

Далі шукаємо вивід порожнього диз'юнкта.

Застосовуємо до першого і третього диз'юнкта правило резолюції:

$$fX_1, X_2, (\neg X_1 \vee X_3), \neg X_3, X_3 g.$$

Застосовуємо до четвертого і п'ятого диз'юнктів правило резолюції:

$$fX_1, X_2, (\neg X_1 \vee X_3), \neg X_3, X_3, \# g.$$

Таким чином порожній диз'юнкт виведений, отже вираз із запереченням висловлювання спростовано, отже саме висловлювання доведено.

Принцип резолюцій повністю замінює аксіому порядку, оскільки вона сама може бути доведена в рамках методу резолюцій.

$$A, B \Rightarrow A;$$

$$A, B, \bar{A} \Rightarrow 0;$$

$$0 < B \Rightarrow 0.$$

Звернемо увагу на те, що посилка B взагалі не використовується, тобто необов'язково треба використовувати всі посилки, головне отримати нуль. Всі доведення клауз починають з приведення їх в нормальну кон'юнктивну форму, потім виписують по порядку всі посилки і склеюють згідно з чергою [8-9].

3.2. Приклади застосування методу резолюцій

1. Метод резолюцій в численні висловлень.

Приклад 1. Задана така множина диз'юнктів S :

$$1) P \vee Q_1 \vee Q_2,$$

$$2) P \vee \overline{Q_3},$$

$$3) U \vee \overline{Q_3},$$

$$4) U \vee \overline{P} \vee \overline{R},$$

$$5) R,$$

$$6) Q_1,$$

7) Q_3 .

Довести методом резолюції, що диз'юнкт U є логічним наслідком з S .

Розв'язання. До множини S додамо диз'юнкт 8) \bar{U} . Тоді з (3) і (8) отримуємо резольвенту 9) \bar{Q}_3 , а з (7) і (9) – резольвенту: 10) \emptyset – порожній диз'юнкт, що і потрібно було довести.

Приклад 2. Методом резолюції перевірити множину S диз'юнктив

$$1) \ p \vee q,$$

$$2) \ p \vee r,$$

$$3) \ \bar{q} \vee \bar{r},$$

$$4) \ \bar{p}$$

на суперечність.

Розв'язання. Обчислюємо і додаємо резольвенти (в дужках вказані № диз'юнктив):

$$5) \ q \ (1, 4);$$

$$6) \ r \ (2, 4);$$

$$7) \ \bar{q} \ (3, 6);$$

$$8) \ \emptyset \ (5, 7).$$

Множина S диз'юнктив є суперечливою.

Приклад 3. Методом резолюції з'ясувати, чи є логічно правильним наступне просте міркування. Студент піде додому (p) або залишиться в університеті (q). Він не залишився в університеті. Отже, студент пішов додому.

Розв'язання. Запишемо це міркування символічно за допомогою вказаних в дужках букв: $p \vee q$, \bar{q} , p . Істинність наслідку визначатиметься істинністю наявних висловів:

$$p \vee q, \bar{q} \vdash p.$$

Застосуємо принцип дедукції:

$$p \vee q, \bar{q}, \bar{p} \vdash \emptyset.$$

Суперечність множини диз'юнктив доведемо за допомогою методу резолюції:

- 1) $p \vee q$;
- 2) \bar{q} ;
- 3) \bar{p} ;
- 4) $p(1, 2)$;
- 5) $\emptyset(3, 4)$.

Тобто дане міркування є логічно правильним.

2. Метод резолюцій в численні предикатів.

Приклад 4. Згідно принципу Пітера, «службовець підвищується по службі то тих пір, поки він не досягне свого рівня компетенції». Чи випливає з цього, що «не існує компетентних начальників»?

Розв'язання. Перевіримо цей висновок за допомогою методу резолюцій. Формалізуємо задачу. Введемо предикати: $C(x)$: « x – службовець», $K(x)$: « x – компетентний», $N(x)$: « x – начальник», $P(x)$: « x підвищується по службі». У якості першої посылки беремо твердження про те, що компетентний службовець підвищується по службі:

$$\forall x(C(x) \wedge K(x) \rightarrow P(x)) .$$

Друга посылка – це твердження про те, що начальник не підвищується по службі:

$$\forall x(N(x) \rightarrow \bar{P}(x)) .$$

Врахуємо також той факт, що начальник є теж службовцем:

$$\forall x(N(x) \rightarrow C(x)) .$$

Тоді висновок, який необхідно перевірити, можна сформулювати так: «всі начальники некомпетентні»:

$$\forall x(N(x) \rightarrow \bar{K}(x)) .$$

Запишемо посылки і заперечення висновку у сколемівській нормальній формі та побудуємо резолютивний вивід:

- 1) $N(a)$;

- 2) $K(a)$;
- 3) $\bar{C}(x) \vee \bar{K}(x) \vee P(x)$;
- 4) $\bar{N}(x) \vee \bar{P}(x)$;
- 5) $\bar{N}(x) \vee C(x)$;
- 6) $\bar{C}(a) \vee P(a)$ ($\{a/x\}$ в 3, резольвента 2, 3);
- 7) $\bar{N}(a) \vee P(a)$ ($\{a/x\}$ в 5, резольвента 5, 6);
- 8) $\bar{N}(a)$ ($\{a/x\}$ в 4, резольвента 4, 7);
- 9) \emptyset (резольвента 1, 8).

Таким чином, компетентних начальників не має.

Приклад 5. Довести, що з посилки «Студенти є громадянами» випливає висновок «Голоси студентів є голосами громадян».

Розв'язання. Нехай $S(x)$: « x – студент», $C(x)$: « x – громадянин» і $V(x, y)$: « x є голос y ». Тоді $\forall y(S(y) \rightarrow C(y))$ – це посилка, $\forall x(\exists y(S(y) \wedge V(x, y)) \rightarrow \exists z(C(z) \wedge V(x, z)))$ – висновок. Сколемівська нормальна форма посилки така:

$$\forall y(\bar{S}(y) \vee C(y)).$$

Знайдемо сколемівську нормальну форму (СНФ) заперечення висновку. Оскільки

$$\begin{aligned} & \overline{\forall x(\exists y(S(y) \wedge V(x, y)) \rightarrow \exists z(C(z) \wedge V(x, z)))} = \\ & = \overline{\forall x(\forall y(\bar{S}(y) \vee \bar{V}(x, y)) \vee \exists z(C(z) \wedge V(x, z)))} = \\ & = \exists x \exists y \forall z((S(y) \wedge V(x, y)) \wedge (\bar{C}(z) \vee \bar{V}(x, z))), \end{aligned}$$

то СНФ така:

$$\forall z((S(b) \wedge V(a, b)) \wedge (\bar{C}(z) \vee \bar{V}(a, z))).$$

Запишемо множину диз'юнктивів і побудуємо резолютивний вивід:

- 1) $S(b)$;
- 2) $\bar{S}(y) \vee C(y)$;
- 3) $V(a, b)$;
- 4) $\bar{C}(z) \vee \bar{V}(a, z)$;

5) $C(b)$ ($\{b/y\}$ в 2, резольвента 1, 2);

6) $\bar{V}(a,b)$ ($\{b/z\}$ в 4, резольвента 4, 5);

7) \emptyset (резольвента 3, 6).

Твердження доведено [10].

3.3. Алгоритмізація за темою роботи

На головній сторінці тренажера відображається інформацію щодо його назви, розробника і керівника. Користувач може переглянути теоретичний матеріал, приклади або відразу перейти до проходження. Тренажер містить декілька прикладів з використання методу резолюцій в численні висловлень і численні предикатів. На кожному кроці виводиться умова, завдання та наводяться варіанти відповіді. Розглянемо алгоритм кожного прикладу.

Приклад 1.

Крок 1. Відображається умова прикладу: «Задана така множина диз'юнктивів S :

$$1) P \vee Q_1 \vee Q_2,$$

$$2) P \vee \bar{Q}_3,$$

$$3) U \vee \bar{Q}_3,$$

$$4) U \vee \bar{P} \vee \bar{R},$$

$$5) R,$$

$$6) Q_1,$$

$$7) Q_3.$$

Довести методом резолюції, що диз'юнкт U є логічним наслідком з S . Користувачу пропонується розв'язати даний приклад або завершити тренінг.

Якщо вибрано розв'язання прикладу, то відбувається перехід на наступний крок.

Крок 2. Відображається завдання: «До множини S додамо диз'юнкт 8) \overline{U} . З яких диз'юнктів отримується резольвента 9) $\overline{Q_3}$? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- (3) і (8);
- (3) і (4);
- (2) і (7);
- (2) і (8).

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 3, інакше – повідомлення про помилку: «З диз'юнктів (3) і (8) отримується резольвента 9) $\overline{Q_3}$ ».

Крок 3. Відображається завдання: «До множини S додамо диз'юнкт 8) \overline{U} . Тоді з (3) і (8) отримуємо резольвенту 9) $\overline{Q_3}$. Чи доведено, що диз'юнкт U є логічним наслідком з S ? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- Так, диз'юнкт U є логічним наслідком з S ;
- Ні, диз'юнкт U не є логічним наслідком з S ;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 4, інакше – повідомлення про помилку: «Потрібно продовжити доведення!».

Крок 4. Відображається завдання: «До множини S додамо диз'юнкт 8) \overline{U} . Тоді з (3) і (8) отримуємо резольвенту 9) $\overline{Q_3}$. Яку резольвенту отримуємо з (7) і (9)? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- 10) Q_3 ;
- 10) $\overline{Q_3}$;
- 10) \emptyset ;
- 10) U .

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 5, інакше – повідомлення про помилку: «З диз'юнктів (7) і (9) отримується резольвента 10) \emptyset – порожній диз'юнкт».

Крок 5. Відображається завдання: «До множини S додамо диз'юнкт 8) \bar{U} . Тоді з (3) і (8) отримуємо резольвенту 9) \bar{Q}_3 , а з (7) і (9) – резольвенту: 10) \emptyset – порожній диз'юнкт. Чи доведено, що диз'юнкт U є логічним наслідком з S ? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- Так, диз'юнкт U є логічним наслідком з S ;
- Ні, диз'юнкт U не є логічним наслідком з S ;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 6, інакше – повідомлення про помилку: «Методом резолюції доведено, що диз'юнкт U є логічним наслідком з S ».

Крок 6. Відображається повідомлення, що завершено виконання прикладу та пропонується перейти до наступного прикладу або завершити проходження тренажера.

Приклад 2.

Крок 1. Відображається умова прикладу: «Методом резолюції перевірити множину S диз'юнктив

- 1) $p \vee q$,
- 2) $p \vee r$,
- 3) $\bar{q} \vee \bar{r}$,
- 4) \bar{p}

на суперечність». Користувачу пропонується розв'язати даний приклад або завершити тренінг.

Якщо вибрано розв'язання прикладу, то відбувається перехід на наступний крок.

Крок 2. Відображається завдання: «З яких диз'юнктив отримується резольвента 5) q ? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- (1) і (3);
- (1) і (4);
- (2) і (3);
- (2) і (4).

Якщо користувач вибирає другий варіант, то відбувається перехід на крок 3, інакше – повідомлення про помилку: «З диз'юнктив (1) і (4) отримується резольвента 5) q ».

Крок 3. Відображається завдання: «З диз'юнктив (1) і (4) отримується резольвента 5) q . Чи доведено, що множина S диз'юнктив є суперечливою? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- Так, множина S диз'юнктив є суперечливою;
- Ні, множина S диз'юнктив не є суперечливою;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 4, інакше – повідомлення про помилку: «Потрібно продовжити доведення!».

Крок 4. Відображається завдання: «З диз'юнктив (1) і (4) отримується резольвента 5) q . Яку резольвенту отримуємо з (2) і (4)? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- б) p ;
- б) \bar{p} ;
- б) \bar{r} ;
- б) r .

Якщо користувач вибирає четвертий варіант, то відбувається перехід на крок 5, інакше – повідомлення про помилку: «З диз'юнктів (2) і (4) отримується резольвента 6) r ».

Крок 5. Відображається завдання: «З диз'юнктів (1) і (4) отримується резольвента 5) q , з (2) і (4) отримується 6) r . Чи доведено, що множина S диз'юнктів є суперечливою? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- Так, множина S диз'юнктів є суперечливою;
- Ні, множина S диз'юнктів не є суперечливою;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 6, інакше – повідомлення про помилку: «Потрібно продовжити доведення!».

Крок 6. Відображається завдання: «З диз'юнктів (1) і (4) отримується резольвента 5) q , з (2) і (4) отримується 6) r . З яких диз'юнктів отримується резольвента 7) \bar{q} ? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- (3) і (5);
- (3) і (6);
- (1) і (5);
- (2) і (6).

Якщо користувач вибирає другий варіант, то відбувається перехід на крок 7, інакше – повідомлення про помилку: «З диз'юнктів (3) і (6) отримується резольвента 7) \bar{q} ».

Крок 7. Відображається завдання: «З диз'юнктів (1) і (4) отримується резольвента 5) q , з (2) і (4) отримується 6) r , з (3) і (6) отримується 7) \bar{q} . Чи доведено, що множина S диз'юнктів є суперечливою? Виберіть правильну відповідь.».

Наводяться варіанти відповіді:

- Так, множина S диз'юнктив є суперечливою;
- Ні, множина S диз'юнктив не є суперечливою;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 8, інакше – повідомлення про помилку: «Потрібно продовжити доведення!».

Крок 8. Відображається завдання: «З диз'юнктив (1) і (4) отримується резольвента 5) q , з (2) і (4) отримується 6) r , з (3) і (6) отримується 7) \bar{q} . Яку резольвенту отримуємо з (5) і (7)? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- 8) q ;
- 8) \bar{q} ;
- 8) r ;
- 8) \emptyset .

Якщо користувач вибирає четвертий варіант, то відбувається перехід на крок 9, інакше – повідомлення про помилку: «З диз'юнктив (5) і (7) отримується резольвента 8) \emptyset ».

Крок 9. Відображається завдання: «З диз'юнктив (1) і (4) отримується резольвента 5) q , з (2) і (4) отримується 6) r , з (3) і (6) отримується 7) \bar{q} , з (5) і (7) отримується 8) \emptyset . Чи доведено, що множина S диз'юнктив є суперечливою? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- Так, множина S диз'юнктив є суперечливою;
- Ні, множина S диз'юнктив не є суперечливою;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 10, інакше – повідомлення про помилку: «Методом резолюції доведено, що множина S диз'юнктив є суперечливою».

Крок 10. Відображається повідомлення, що завершено виконання прикладу та пропонується перейти до наступного прикладу або завершити проходження тренажера.

Приклад 3.

Крок 1. Відображається умова прикладу: «Методом резолюції з'ясувати, чи є логічно правильним наступне просте міркування. Студент піде додому (p) або залишиться в університеті (q). Він не залишився в університеті. Отже, студент пішов додому». Користувачу пропонується розв'язати даний приклад або завершити тренінг.

Якщо вибрано розв'язання прикладу, то відбувається перехід на наступний крок.

Крок 2. Відображається завдання: «Запишемо це міркування символічно за допомогою вказаних в дужках букв: $p \vee q$, \bar{q} , p . Істинність наслідку визначатиметься істинністю наявних висловів: $p \vee q$, $\bar{q} \vdash p$.

Застосуємо принцип дедукції: $p \vee q$, \bar{q} , $\bar{p} \vdash \emptyset$.

Суперечність множини диз'юнктив доведемо за допомогою методу резолюції. Визначіть множину диз'юнктив. Виберіть правильні відповіді». Наводяться варіанти відповіді:

- $p \vee q$;
- \bar{q} ;
- \bar{p} ;
- \emptyset .

Якщо користувач вибирає перший, другий і третій варіанти, то відбувається перехід на крок 3, інакше – повідомлення про помилку: «Множина диз'юнктив: 1) $p \vee q$, 2) \bar{q} , 3) \bar{p} ».

Крок 3. Відображається завдання: «Суперечність множини диз'юнктив доведемо за допомогою методу резолюції: 1) $p \vee q$, 2) \bar{q} , 3) \bar{p} . З яких диз'юнктив отримується резольвента 4) p ? Виберіть правильну відповідь».

Наводяться варіанти відповіді:

- (1) і (2);
- (1) і (3);
- (2) і (3).

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 4, інакше – повідомлення про помилку: «З диз'юнктивів (1) і (2) отримується резольвента 4) p ».

Крок 4. Відображається завдання: «Суперечність множини диз'юнктивів доведемо за допомогою методу резолюції: 1) $p \vee q$, 2) \bar{q} , 3) \bar{p} . З диз'юнктивів (1) і (2) отримується резольвента 4) p . Чи доведено, що міркування є логічно правильним? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- Так, дане міркування є логічно правильним;
- Ні, дане міркування не є логічно правильним;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 5, інакше – повідомлення про помилку: «Потрібно продовжити доведення!».

Крок 5. Відображається завдання: «Суперечність множини диз'юнктивів доведемо за допомогою методу резолюції: 1) $p \vee q$, 2) \bar{q} , 3) \bar{p} . З диз'юнктивів (1) і (2) отримується резольвента 4) p . Яку резольвенту отримуємо з (3) і (4)? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- 5) \bar{p} ;
- 5) p ;
- 5) \emptyset .

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 6, інакше – повідомлення про помилку: «З диз'юнктивів (3) і (4) отримується резольвента 5) \emptyset – порожній диз'юнкт».

Крок 6. Відображається завдання: «Суперечність множини диз'юнктивів доведемо за допомогою методу резолюції: 1) $p \vee q$, 2) \bar{q} , 3) \bar{p} . З диз'юнктивів (1) і (2) отримується резольвента 4) p , з (3) і (4) отримується 5) \emptyset – порожній диз'юнкт. Чи доведено, що міркування є логічно правильним? Виберіть правильну відповідь.». Наводяться варіанти відповіді:

- Так, дане міркування є логічно правильним;
- Ні, дане міркування не є логічно правильним;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 7, інакше – повідомлення про помилку: «Методом резолюції доведено, що дане міркування є логічно правильним».

Крок 7. Відображається повідомлення, що завершено виконання прикладу та пропонується перейти до наступного прикладу або завершити проходження тренажера.

Приклад 4.

Крок 1. Відображається умова прикладу: «Згідно принципу Пітера, «службовець підвищується по службі то тих пір, поки він не досягне свого рівня компетенції». Чи впливає з цього, що «не існує компетентних начальників»?». Користувачу пропонується розв'язати даний приклад або завершити тренінг.

Якщо вибрано розв'язання прикладу, то відбувається перехід на наступний крок.

Крок 2. Відображається завдання: «Перевіримо цей висновок за допомогою методу резолюцій. Формалізуємо задачу. Введемо предикати: $S(x)$: « x – службовець», $K(x)$: « x – компетентний», $N(x)$: « x – начальник», $P(x)$: « x підвищується по службі». У якості першої посилки сформулюйте твердження про те, що компетентний службовець підвищується по службі. Виберіть правильну відповідь».

Наводяться варіанти відповіді:

- $\forall x(C(x) \wedge K(x) \wedge P(x));$
- $\forall x(C(x) \rightarrow K(x) \rightarrow P(x));$
- $\forall x(C(x) \wedge K(x) \rightarrow P(x));$
- $\forall x(C(x) \rightarrow K(x) \wedge P(x)).$

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 3, інакше – повідомлення про помилку: «У якості першої посилки беремо твердження про те, що компетентний службовець підвищується по службі: $\forall x(C(x) \wedge K(x) \rightarrow P(x))$ ».

Крок 3. Відображається завдання: «У якості першої посилки беремо твердження про те, що компетентний службовець підвищується по службі: $\forall x(C(x) \wedge K(x) \rightarrow P(x))$.

Друга посилка – це твердження про те, що начальник не підвищується по службі. Виберіть правильну відповідь». Наводяться варіанти відповіді:

- $\forall x(N(x) \rightarrow \bar{P}(x));$
- $\forall x(N(x) \vee \bar{P}(x));$
- $\forall x(N(x) \wedge \bar{P}(x));$
- $\exists x(N(x) \rightarrow \bar{P}(x)).$

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 4, інакше – повідомлення про помилку: «Друга посилка – це твердження про те, що начальник не підвищується по службі: $\forall x(N(x) \rightarrow \bar{P}(x))$ ».

Крок 4. Відображається завдання: «Друга посилка – це твердження про те, що начальник не підвищується по службі: $\forall x(N(x) \rightarrow \bar{P}(x))$. Врахуємо також той факт, що начальник є теж службовцем. Виберіть правильну відповідь». Наводяться варіанти відповіді:

- $\exists x(N(x) \rightarrow C(x));$

- $\forall x(N(x) \rightarrow C(x))$;
- $\forall x(N(x) \vee C(x))$;
- $\forall x(N(x) \wedge C(x))$.

Якщо користувач вибирає другий варіант, то відбувається перехід на крок 5, інакше – повідомлення про помилку: «Врахуємо також той факт, що начальник є теж службовцем: $\forall x(N(x) \rightarrow C(x))$ ».

Крок 5. Відображається завдання: «Врахуємо також той факт, що начальник є теж службовцем: $\forall x(N(x) \rightarrow C(x))$. Тоді висновок, який необхідно перевірити, можна сформулювати так: «всі начальники некомпетентні». Виберіть правильну відповідь». Наводяться варіанти відповіді:

- $\exists x(N(x) \rightarrow \bar{K}(x))$;
- $\forall x(N(x) \wedge \bar{K}(x))$;
- $\forall x(N(x) \vee \bar{K}(x))$;
- $\forall x(N(x) \rightarrow \bar{K}(x))$.

Якщо користувач вибирає четвертий варіант, то відбувається перехід на крок 6, інакше – повідомлення про помилку: «Тоді висновок, який необхідно перевірити, можна сформулювати так: «всі начальники некомпетентні»: $\forall x(N(x) \rightarrow \bar{K}(x))$ ».

Крок 6. Відображається завдання: «Тоді висновок, який необхідно перевірити, можна сформулювати так: «всі начальники некомпетентні»: $\forall x(N(x) \rightarrow \bar{K}(x))$. Чи доведено, що компетентних начальників не має? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- Так, компетентних начальників не має;
- Ні, компетентні начальники існують;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 7, інакше – повідомлення про помилку: «Потрібно продовжити доведення!».

Крок 7. Відображається завдання: «Запишемо посилки і заперечення висновку у сколемівській нормальній формі та побудуємо резолютивний вивід: 1) $N(a)$, 2) $K(a)$, 3) $\bar{C}(x) \vee \bar{K}(x) \vee P(x)$, 4) $\bar{N}(x) \vee \bar{P}(x)$, 5) $\bar{N}(x) \vee C(x)$. З яких диз'юнктів отримується резольвента 6) $\bar{C}(a) \vee P(a)$? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- (2) і (3), $\{a/x\}$ в (3);
- (2) і (4), $\{a/x\}$ в (4);
- (2) і (5), $\{a/x\}$ в (5);
- (1) і (4), $\{a/x\}$ в (4).

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 8, інакше – повідомлення про помилку: «З диз'юнктів (2) і (3) отримується резольвента 6) $\bar{C}(a) \vee P(a)$, $\{a/x\}$ в (3)».

Крок 8. Відображається завдання: «З диз'юнктів (2) і (3) отримується резольвента 6) $\bar{C}(a) \vee P(a)$. З яких диз'юнктів отримується резольвента 7) $\bar{N}(a) \vee P(a)$? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- (1) і (5), $\{a/x\}$ в (5);
- (4) і (6), $\{a/x\}$ в (4);
- (5) і (6), $\{a/x\}$ в (5);
- (3) і (6), $\{a/x\}$ в (3).

Якщо користувач вибирає третій варіант, то відбувається перехід на крок 9, інакше – повідомлення про помилку: «З диз'юнктів (5) і (6) отримується резольвента 7) $\bar{N}(a) \vee P(a)$, $\{a/x\}$ в (5)».

Крок 9. Відображається завдання: «З диз'юнктивів (2) і (3) отримується резольвента 6) $\overline{C}(a) \vee P(a)$, з (5) і (6) отримується 7) $\overline{N}(a) \vee P(a)$. З яких диз'юнктивів отримується резольвента 8) $\overline{N}(a)$? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- (1) і (7);
- (4) і (7), $\{a/x\}$ в (4);
- (5) і (7), $\{a/x\}$ в (5);
- (4) і (5).

Якщо користувач вибирає другий варіант, то відбувається перехід на крок 10, інакше – повідомлення про помилку: «З диз'юнктивів (4) і (7) отримується резольвента 8) $\overline{N}(a)$, $\{a/x\}$ в (4)».

Крок 10. Відображається завдання: «З диз'юнктивів (2) і (3) отримується резольвента 6) $\overline{C}(a) \vee P(a)$, з (5) і (6) отримується 7) $\overline{N}(a) \vee P(a)$, з (4) і (7) отримується 8) $\overline{N}(a)$. З яких диз'юнктивів отримується резольвента 9) \emptyset ? Виберіть правильну відповідь». Наводяться варіанти відповіді:

- (1) і (8);
- (1) і (7);
- (4) і (7);
- (5) і (8).

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 11, інакше – повідомлення про помилку: «З диз'юнктивів (1) і (8) отримується резольвента 9) \emptyset ».

Крок 11. Відображається завдання: «З диз'юнктивів (2) і (3) отримується резольвента 6) $\overline{C}(a) \vee P(a)$, з (5) і (6) отримується 7) $\overline{N}(a) \vee P(a)$, з (4) і (7) отримується 8) $\overline{N}(a)$, з (1) і (8) отримується 9) \emptyset . Чи доведено, що компетентних начальників не має? Виберіть правильну відповідь».

Наводяться варіанти відповіді:

- Так, компетентних начальників не має;
- Ні, компетентні начальники існують;
- Ні, потрібно продовжити доведення.

Якщо користувач вибирає перший варіант, то відбувається перехід на крок 12, інакше – повідомлення про помилку: «Методом резолюції доведено, що компетентних начальників не має».

Крок 12. Відображається повідомлення, що завершено виконання прикладу та пропонується пройти тренажер спочатку або завершити проходження [11].

3.4. Розробка блок-схеми

На рисунках 3.1 – 3.2 зображено блок-схему алгоритму роботи тренажеру.

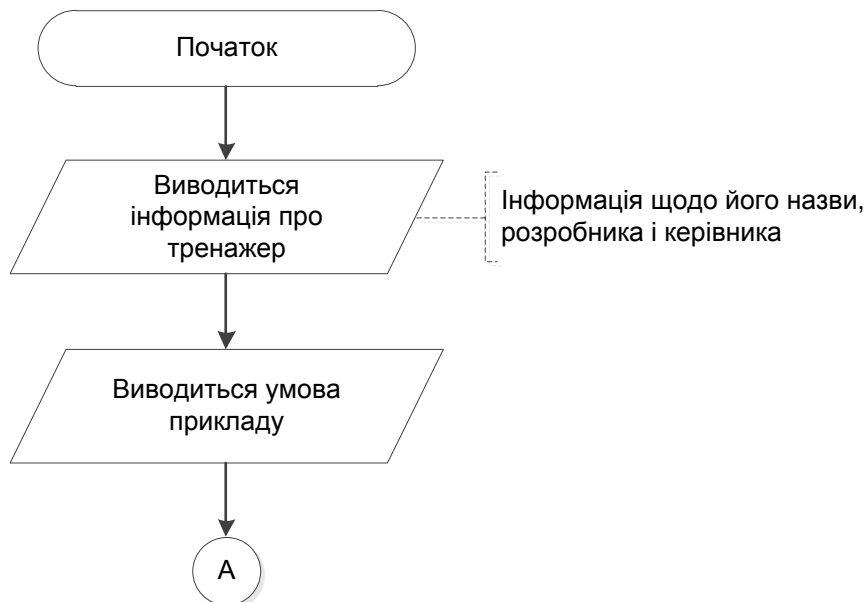


Рисунок 3.1 – Блок-схема алгоритму роботи тренажеру

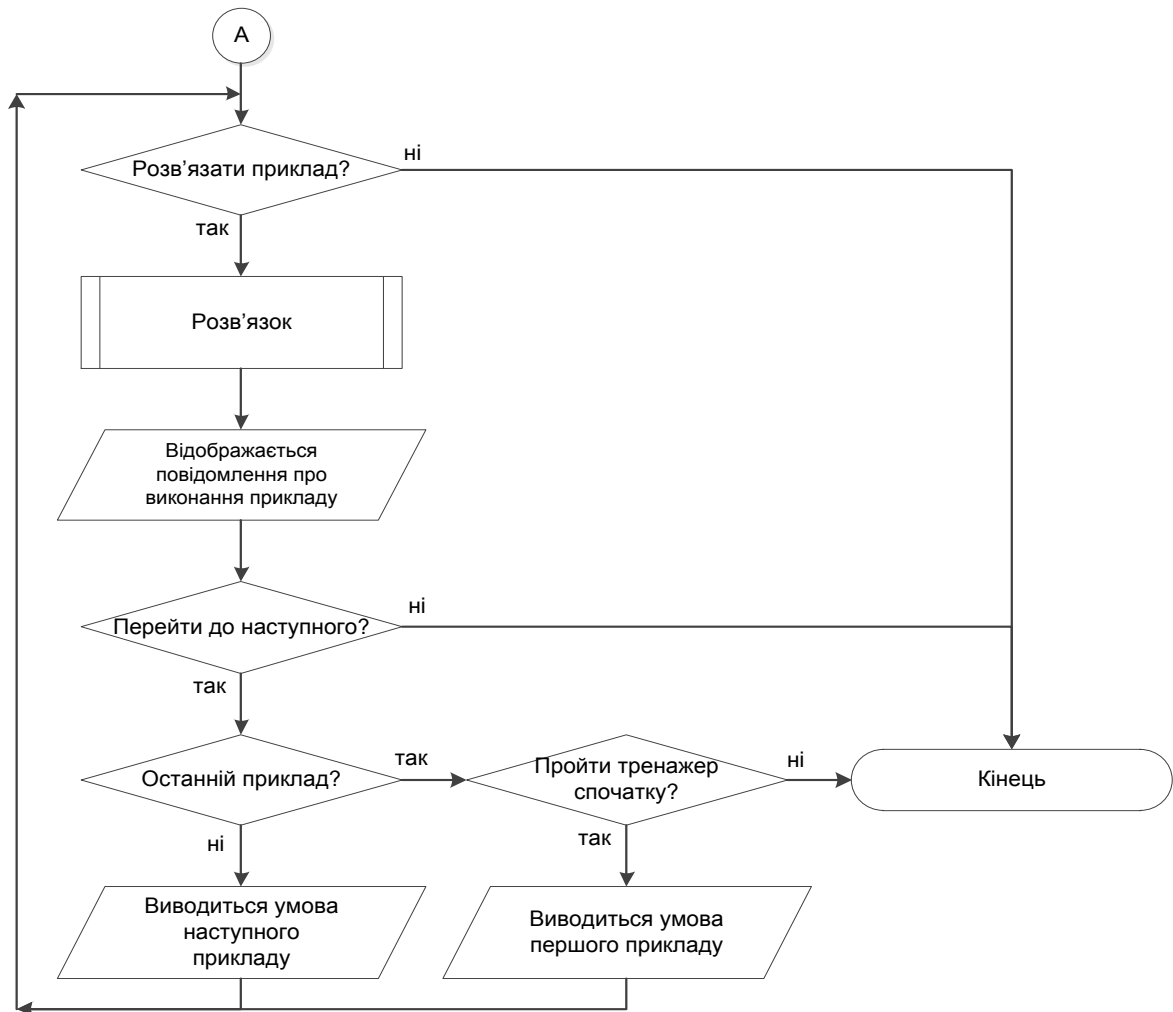


Рисунок 3.2 – Продовження блок-схеми алгоритму роботи тренажеру

На рисунках 3.3-3.4 зображена блок-схема блоку «Розв'язок» для першого прикладу.

На рисунках 3.5-3.7 зображена блок-схема блоку «Розв'язок» для другого прикладу.

На рисунках 3.8-3.11 зображена блок-схема блоку «Розв'язок» для третього прикладу.

На рисунках 3.12-3.15 зображена блок-схема блоку «Розв'язок» для четвертого прикладу.

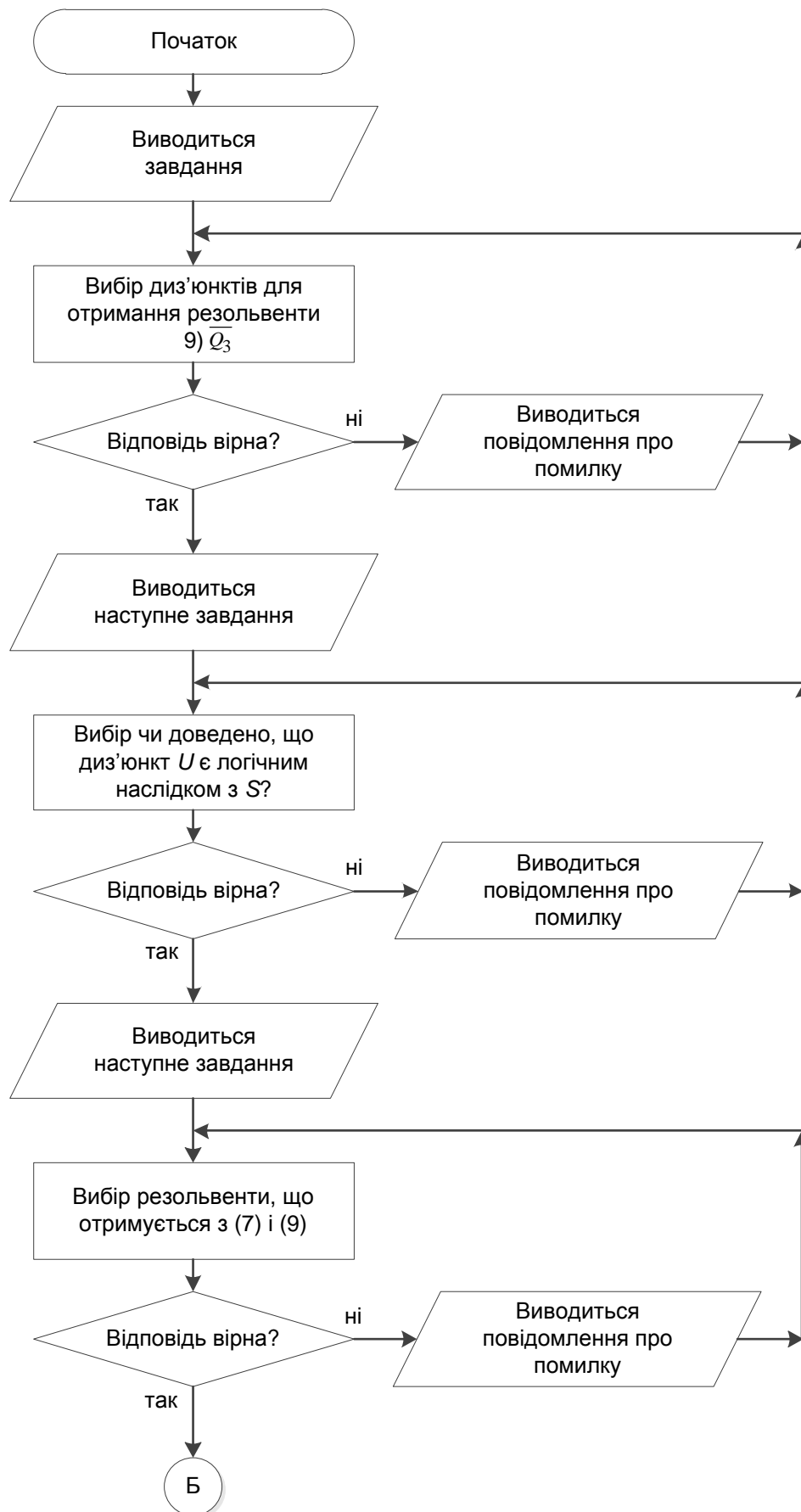


Рисунок 3.3 – Блок-схема блоку «Розв'язок» для першого прикладу

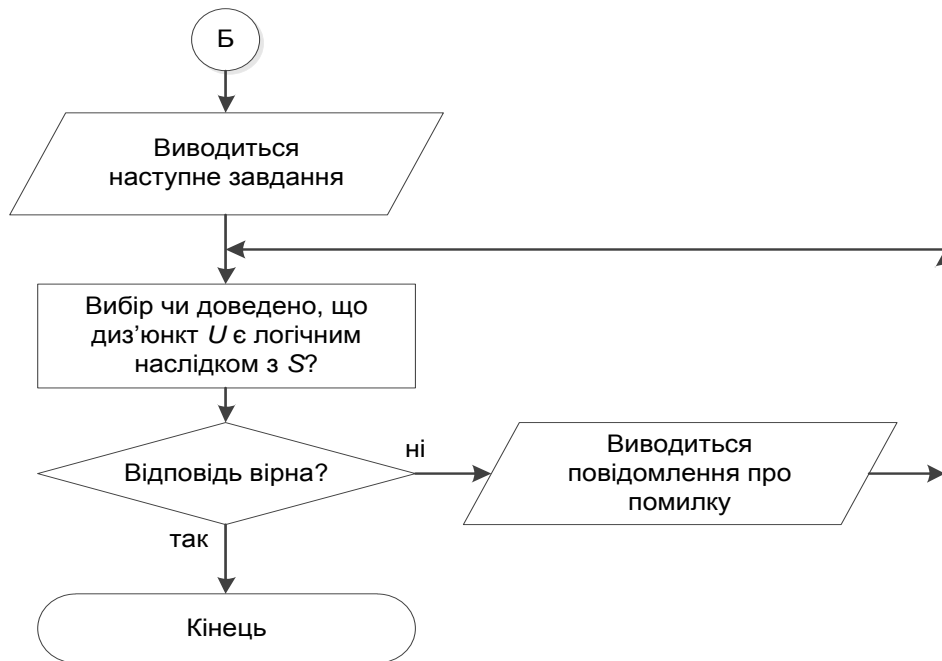


Рисунок 3.4 – Продовження блок-схеми «Розв'язок» для першого прикладу

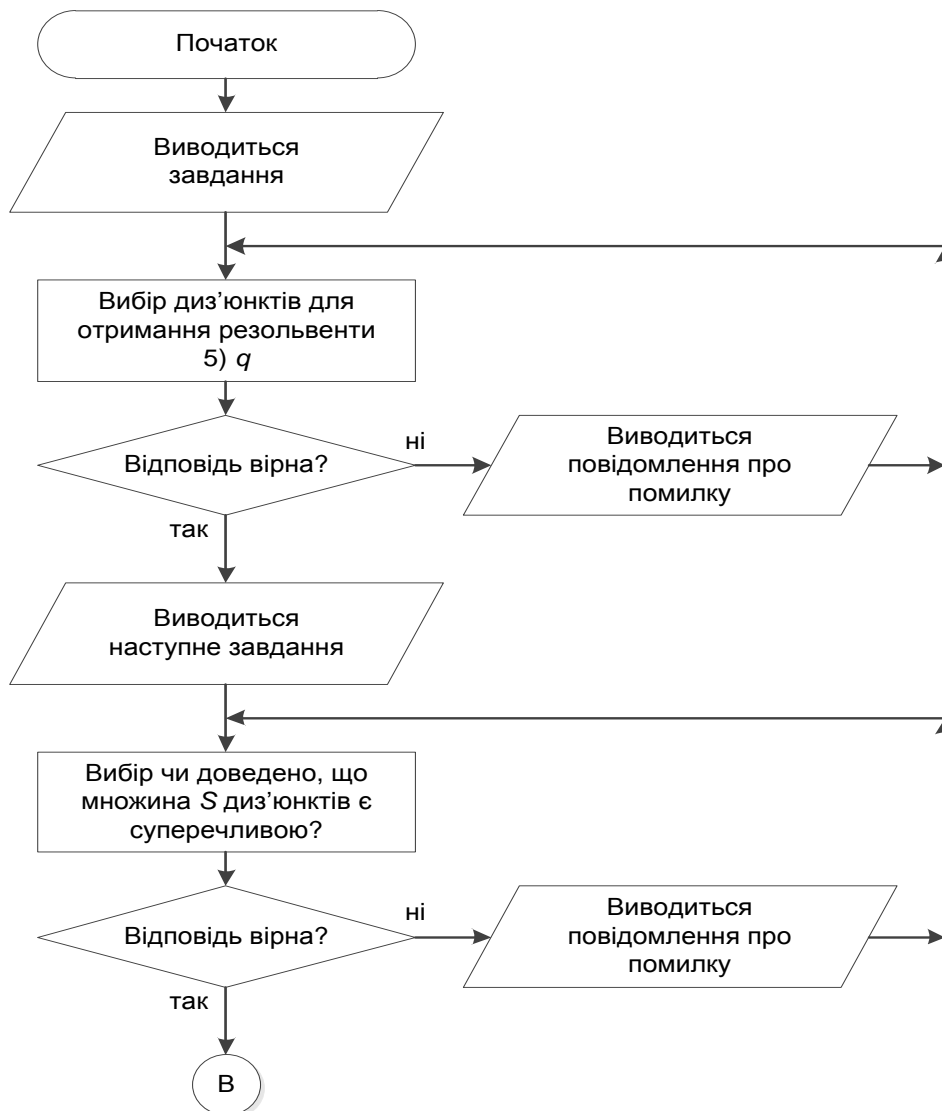


Рисунок 3.5 – Блок-схема блоку «Розв'язок» для другого прикладу

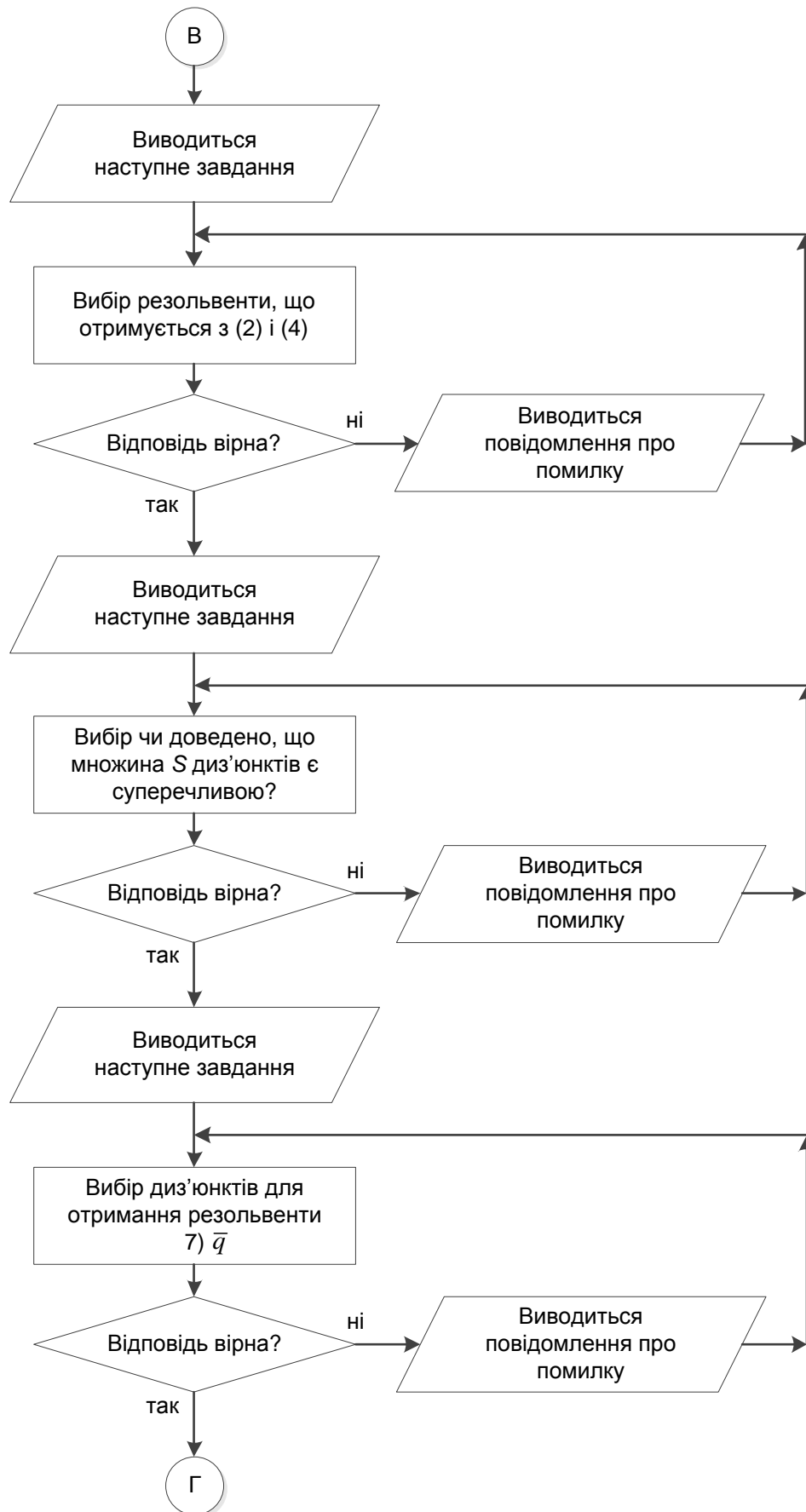


Рисунок 3.6 – Продовження блок-схеми «Розв'язок» для другого прикладу

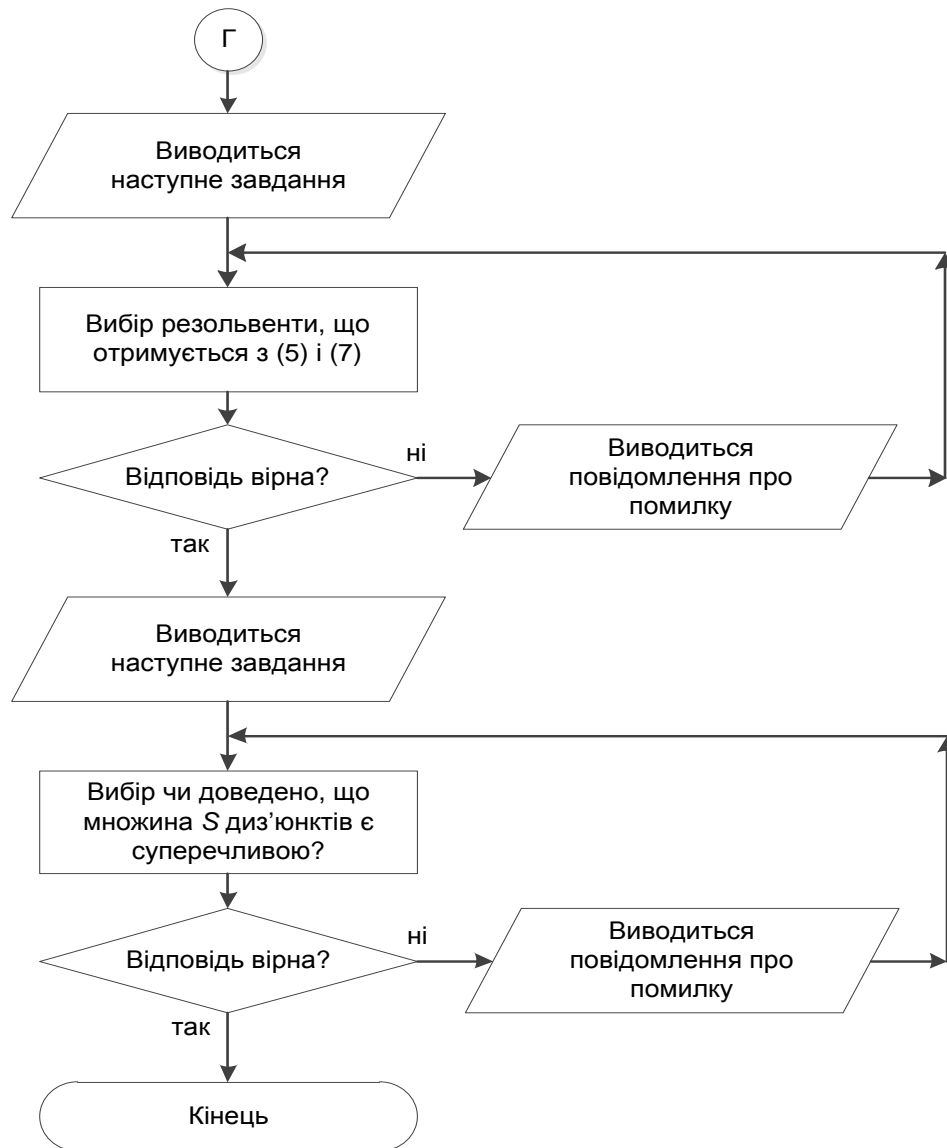


Рисунок 3.7 – Продовження блок-схеми «Розв'язок» для другого прикладу

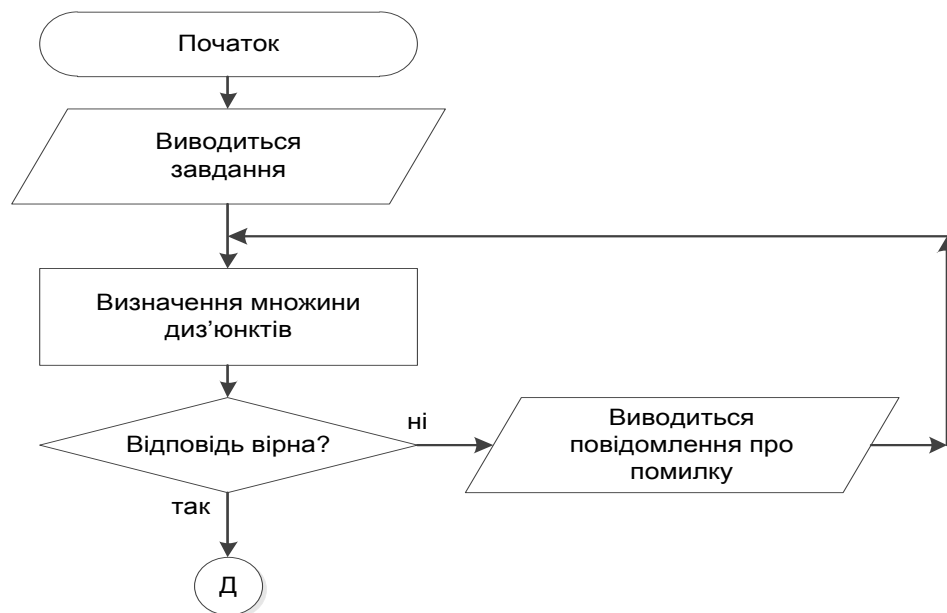


Рисунок 3.8 – Блок-схема блоку «Розв'язок» для третього прикладу

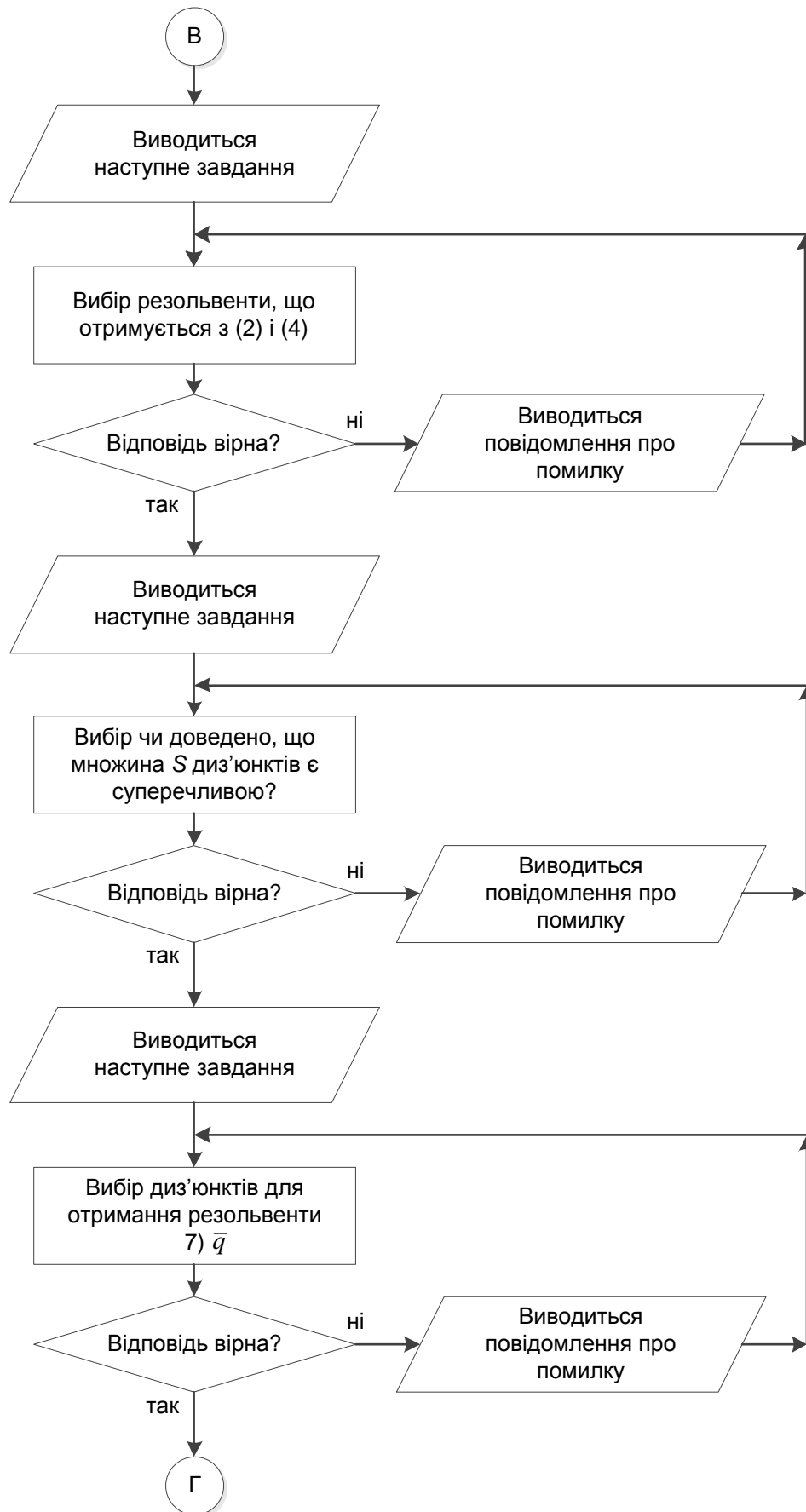


Рисунок 3.9 – Продовження блок-схеми «Розв’язок» для третього прикладу

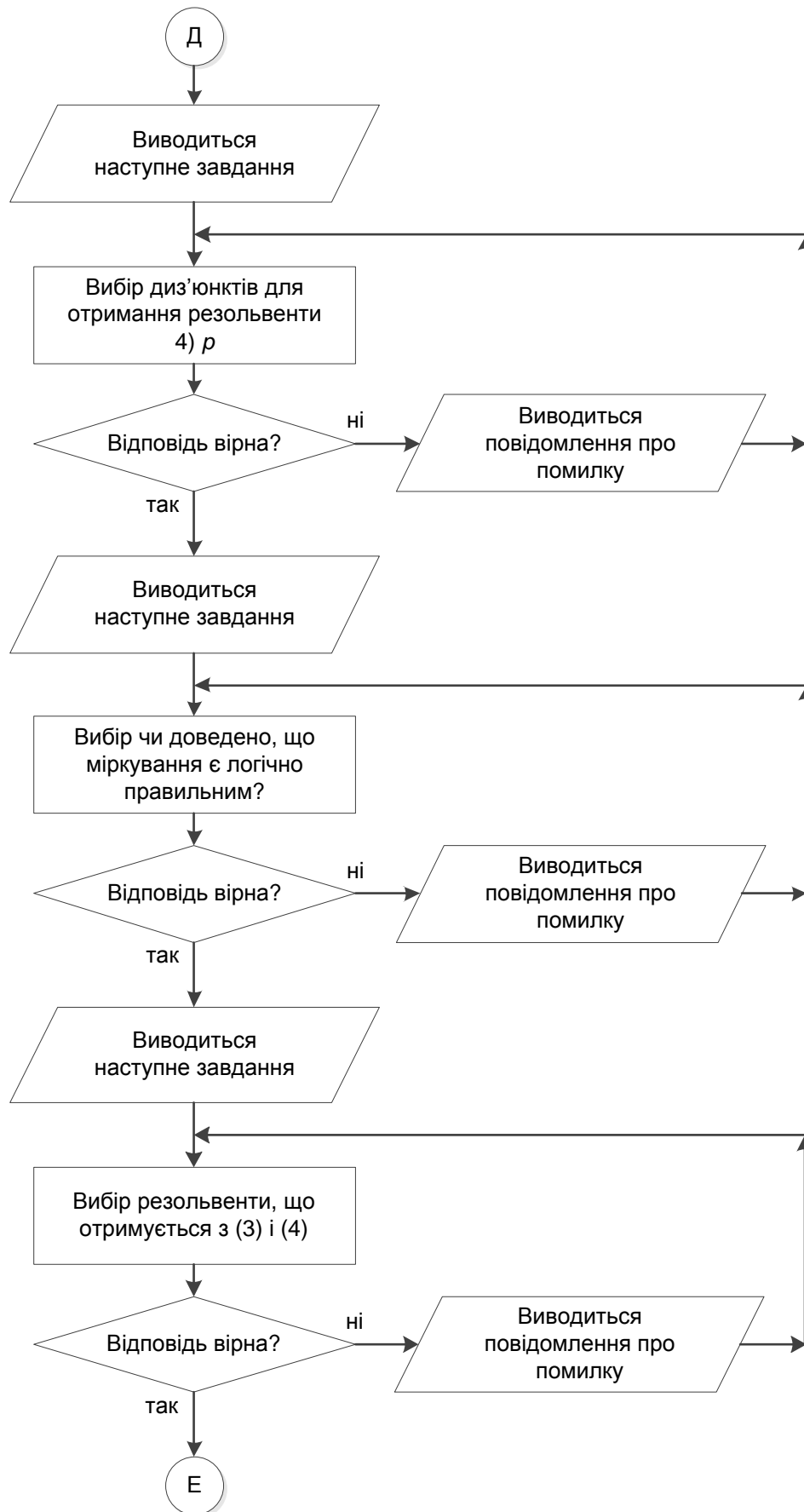


Рисунок 3.10 – Продовження блок-схеми «Розв’язок» для третього прикладу

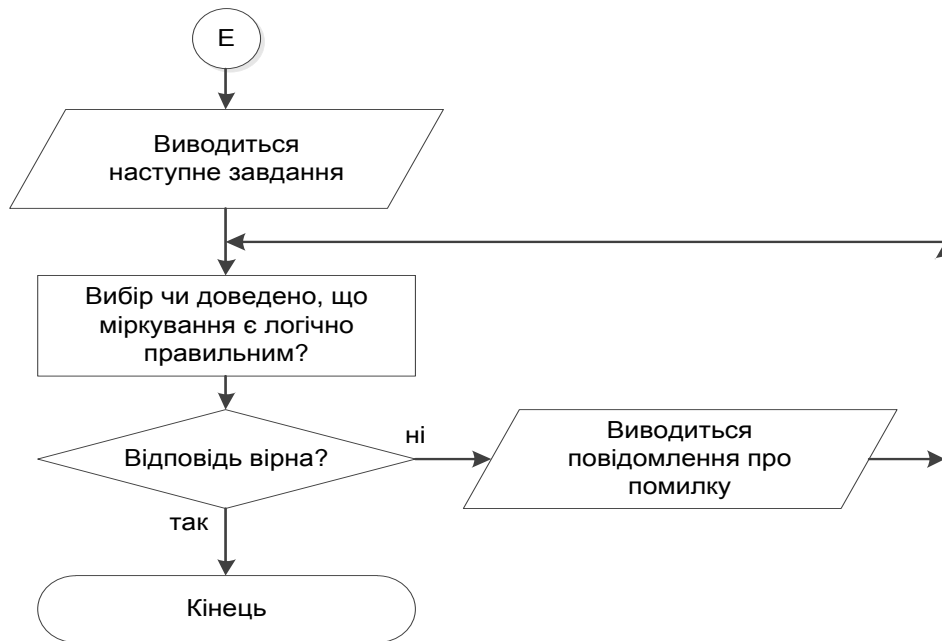


Рисунок 3.11 – Продовження блок-схеми «Розв’язок» для третього прикладу

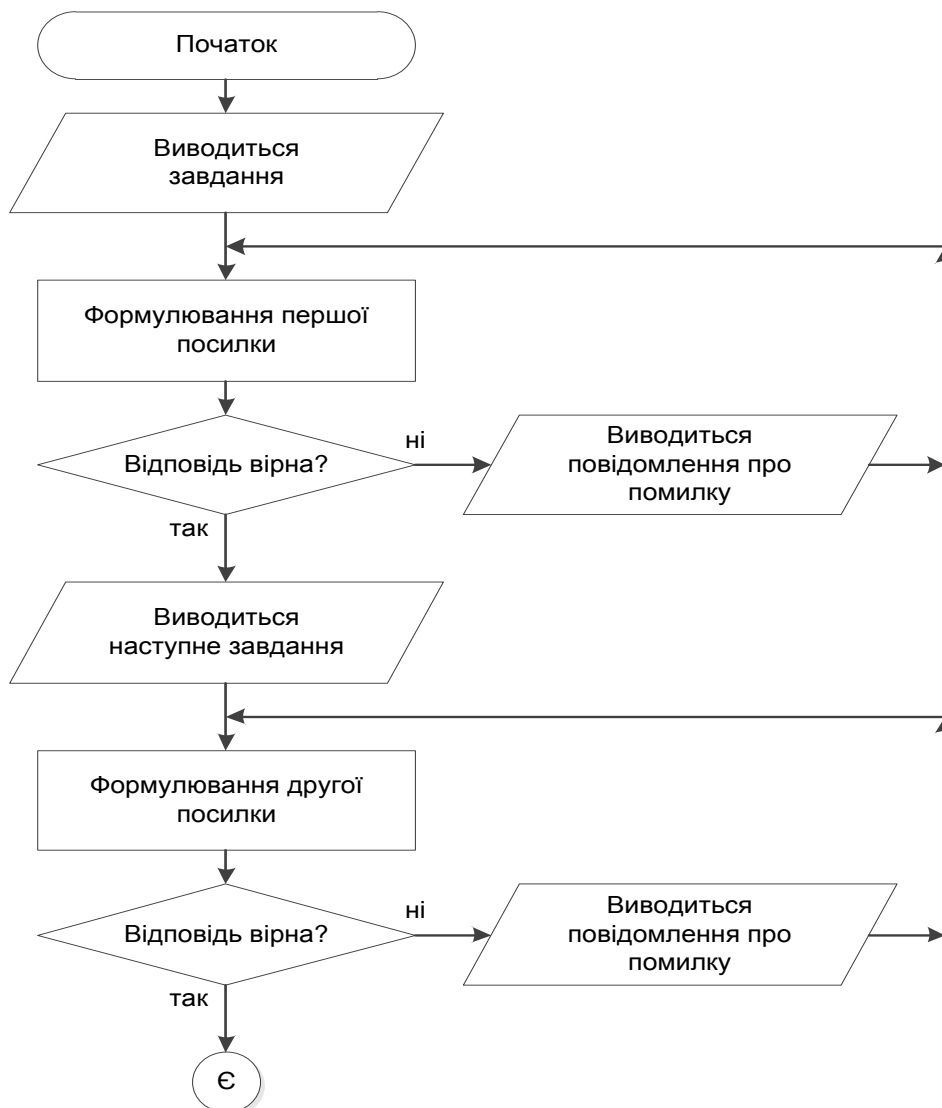


Рисунок 3.12 – Блок-схема блоку «Розв’язок» для четвертого прикладу

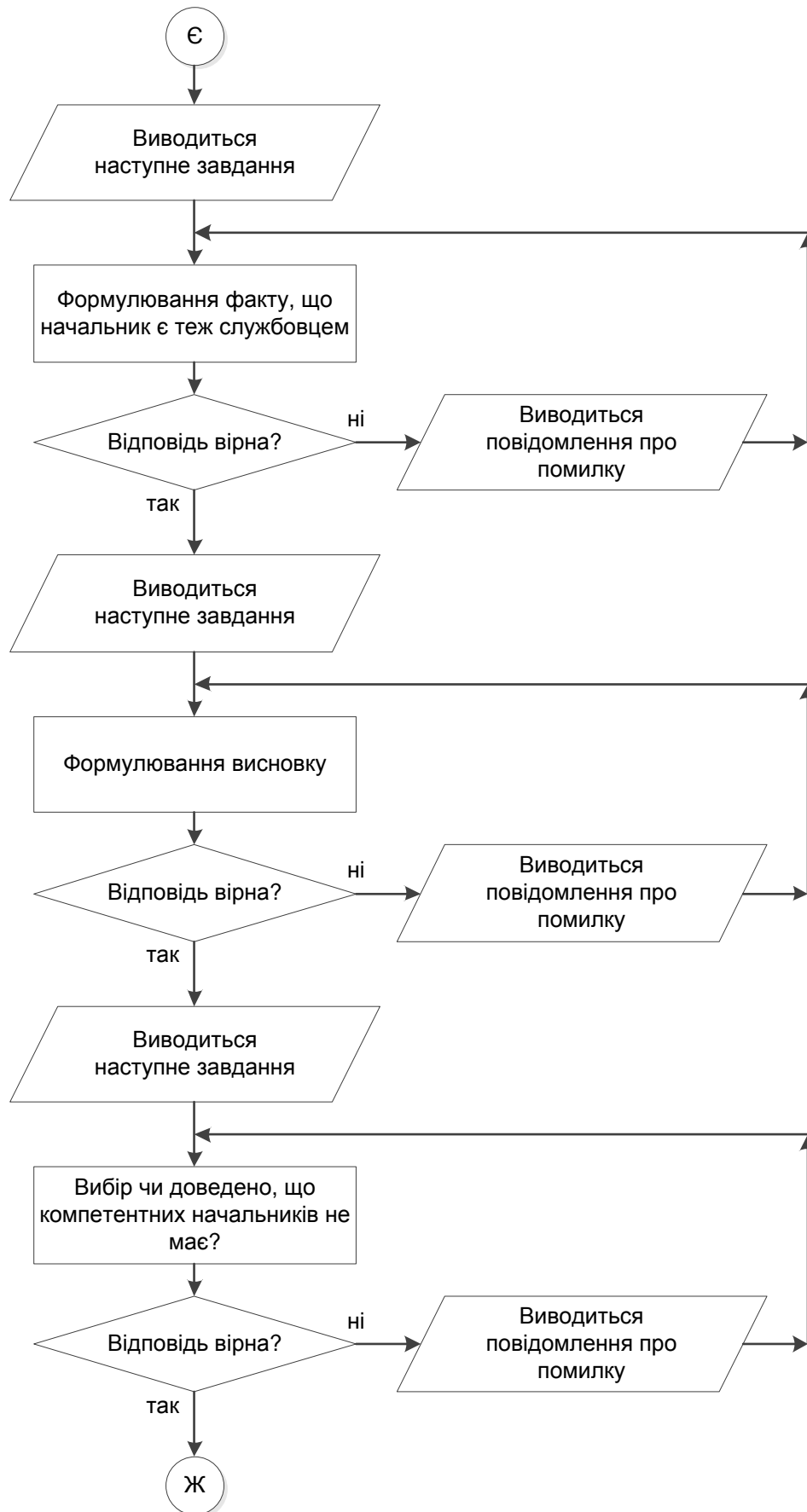


Рисунок 3.13 – Продовження блок-схеми «Розв’язок» для четвертого прикладу

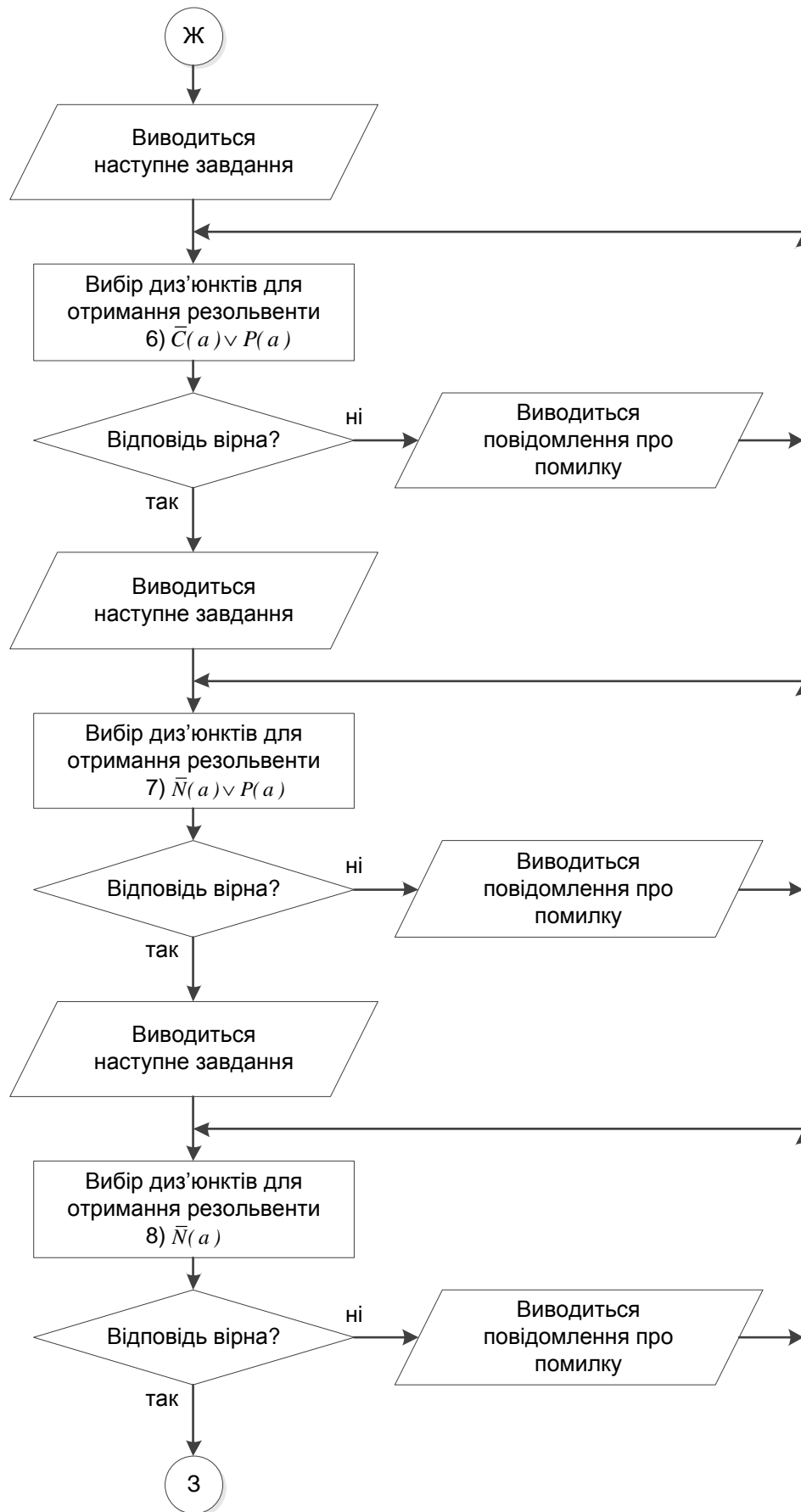


Рисунок 3.14 – Продовження блок-схеми «Розв’язок» для четвертого прикладу

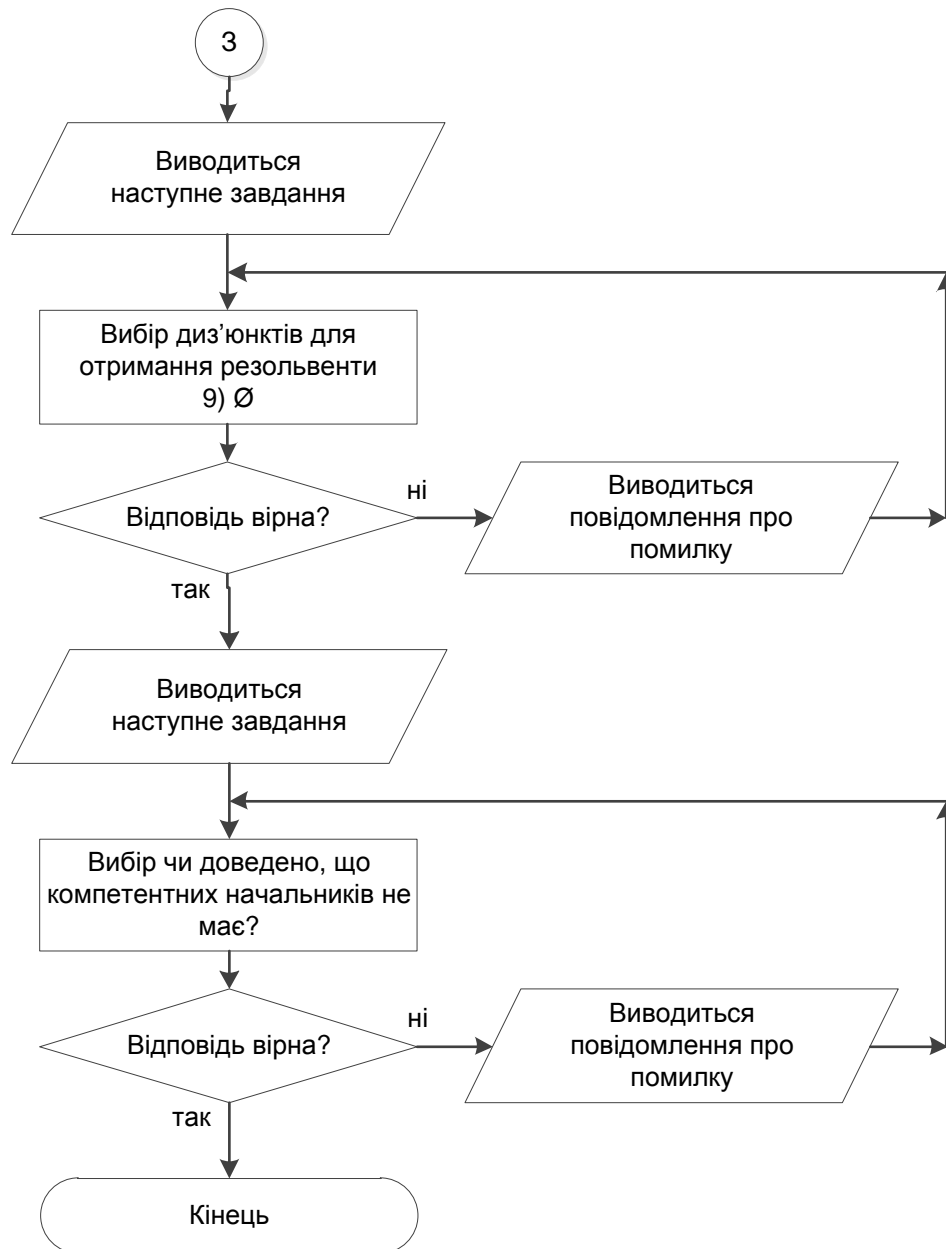


Рисунок 3.15 – Продовження блок-схеми «Розв’язок» для четвертого прикладу

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

Java дозволяє створювати незалежні від платформи програми шляхом компіляції в проміжне представлення, яке називається байткодом. Багато попередніх спроб знати розв'язок проблеми незалежності від платформи були зроблені за рахунок продуктивності. Інтерпретуючі системи, подібні до BASIC, Perl, страждають на майже неподоланий дефіцит продуктивності. Це було враховано при створенні Java. Незважаючи на те, що Java є інтерпретованою мовою, генерація байткодів була ретельно оптимізована в такий спосіб, щоб одержуваний байткод можна було легко перекладати в машинний код, який працює з дуже високою продуктивністю. Виконуючі системи такого роду не втрачають жодних переваг переносимого коду.

Мова Java призначена для створення програм, які працюють в розподіленому середовищі Internet на базі протоколів TCP/IP. Насправді доступ до ресурсів за допомогою URL відрізняється від доступу к файлу. Крім того в Java наявний засіб передачі повідомлень в межах внутрішнього адресного простору. Це дозволяє забезпечити віддалене виконання процедур. Ці інтерфейси включені у пакет RMI (remote method invocation). Цей засіб привносить високий рівень абстракції в програмування для середовища клієнт/сервер.

Java-програми несуть у собі значний обсяг інформації про типи часу виконання (run-time type information), яка використовується для дозволу доступу до об'єктів під час роботи програми. Це дозволяє забезпечити безпечну та оптимальну динамічну компоновку. В такий спосіб досягається захищеність середовища виконання аплетів.

Зазначена вище простота програмування на Java є причиною того, що розробки на Java коштуватимуть дешевше аналогічних на більш

потужних мовах програмування. Цьому ж сприяє і переносимість програм на Java, оскільки ліквідуються витрати пов'язані з адаптацією програми на конкретній платформі.

Програми на Java можуть знайти різне застосування в навчальному процесі: інтерактивні навчаючі програми (HTML в поєднанні з Java), програми-тести і особливо ділові ігри. Додаткові переваги можна отримати, якщо писати ці програми у вигляді аплетів, які ініціалізуються з Web сервера внутрішньої мережі Intranet. В такий спосіб можна уникнути інсталяції програми на багатьох комп'ютерах - користувач просто запускає Web-браузер і загрузає потрібну сторінку.

Поставки інтегрованого середовища розробки NetBeans (integrated development environment, IDE). Середовище IDE NetBeans може підняти вашу продуктивність при роботі з технологіями Java SE, Java EE або Java ME, а також PHP, Groovy, JavaScript і C / C ++. Візуальні засоби, які створюють схему коду, також доступні, що дозволяє створювати базові додатки без написання коду [12].

Ось головні причини для використання середовища IDE NetBeans [13]:

- Відразу готова до роботи;
- Безкоштовно і з відкритим вихідним кодом;
- Потужний GUI Builder;
- Підтримка стандартів і платформ Java;
- Засоби профілювання і налагодження;
- Підтримка динамічної мови;
- Платформа, що розширюється;
- Проекти, які настроюються;
- Підтримка коду, відмінного від Java;
- Спеціальна підтримка.

4.2. Опис процесу програмної реалізації

Робота над тренажером почалась з дизайну. Спочатку розроблено основні панелі:

- StartPanel – стартова сторінка;
- Examples – приклади;
- TheoryPanel – панель з вибором матеріалу;
- TheoryMain – метод резолюцій;
- TheoryExample1 – перший приклад;
- TheoryExample2 – другий приклад;
- TheoryExample3 – третій приклад;
- TheoryExample4 – четвертий приклад.

Потім було підключено бібліотеки і оголошено змінні.

```
import java.awt.CardLayout;
import java.awt.Color;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
```

```
CardLayout cl; //використовується для відображення вказаної
панелі
```

```
int step = 2; //поточний крок алгоритму
```

```
int example = 1; //поточний приклад
```

```
boolean end = false; //вказує, що розв'язано останнє завдання
прикладу.
```

Для відображення повідомлення про помилку створено функцію ShowError(int ex, int i), де параметр int ex – номер прикладу, а int i – номер кроку алгоритму.

```

private void ShowError(int ex, int i) {
    ImageIcon icon;
    if((ex==2 && (i==3 || i==5 || i== 7)) || (ex==3 && i==4) || (ex==4
&& i==6)) {
        icon = new
ImageIcon(getClass().getResource("/image/sol1error3.png"));
    } else {
        icon = new
ImageIcon(getClass().getResource("/image/sol"+ex+"error"+i+".png"));
    }
    JLabel label = new JLabel(icon);
    label.setBackground(Color.white);
    label.setOpaque(false);
    label.setSize(30, 30);
    label.setHorizontalAlignment(SwingConstants.CENTER);
    JOptionPane.showMessageDialog(MainPanel, label, "Помилка",
JOptionPane.ERROR_MESSAGE);
}

```

Щоб вивести умову наступного прикладу реалізовано функцію NextCondition().

```

private void NextCondition() {
    cl = (CardLayout) Conditions.getLayout();
    cl.show(Conditions, "cond"+example);
}

```

Функція NextSolution() перевіряє чи досягнуто останнього кроку алгоритму прикладу. Якщо так, то виводиться розв'язок, якщо ні – наступний крок.

```

private void NextSolution() {
    if(end) {
        cl = (CardLayout) Ends.getLayout();
    }
}

```

```

        cl.show(Ends, "end"+example);
        Examples.setEnabledAt(2, true);
        SNextStep.setEnabled(false);
        Examples.setSelectedIndex(2);
        end=false;
    } else {
        step++;
        if(example==4 && step==7) NextStart(5);
        cl = (CardLayout) Solutions.getLayout();
        cl.show(Solutions, "sol"+example+"step"+step);
    }
}

```

Також було створено функцію NextStart(int start) для зручності зміни умови деяких завдань. В параметрі int start вказується необхідна умова.

```

private void NextStart(int start) {
    cl = (CardLayout) Starts.getLayout();
    cl.show(Starts, "start"+start);
}

```

При виборі мови відбувається подія LanguageItemStateChanged. Вона змінює вміст стартової сторінки і теоретичного матеріалу залежно від вибраної мови (Див. Додаток А).

Якщо натиснути на кнопку «Теорія», то виведеться панель з вибором теоретичного матеріалу завдяки події TheoryActionPerformed. Для кожної кнопки на цій панелі відводиться схожа подія. Так для виведення інформації про метод резолюцій використовується TMainActionPerformed.

```

private void TheoryActionPerformed(java.awt.event.ActionEvent evt) {
    TheoryPanel.setSize(318, 367);
    TheoryPanel.setLocationRelativeTo(MainPanel);
    TheoryPanel.setVisible(true);
}

```

```
private void TMainActionPerformed(java.awt.event.ActionEvent evt) {
    TheoryMain.setSize(670, 630);
    TheoryMain.setLocationRelativeTo(TheoryPanel);
    TheoryMain.setVisible(true);
}
```

Натиснувши «Завершити перегляд» завдяки TExitMouseClicked панель з теоретичним матеріалом закриється.

```
private void TExitMouseClicked(java.awt.event.MouseEvent evt) {
    TheoryPanel.setVisible(false);
}
```

При натисненні кнопки «Практика» відбудеться подія PracticeActionPerformed, що відобразить умову прикладу.

```
private void PracticeActionPerformed(java.awt.event.ActionEvent evt)
{
    if(Language.getSelectedIndex()==1) {
        Method_Of_Resolutions.StartAppletEN();
    } else {
        cl = (CardLayout) MainPanel.getLayout();
        cl.show(MainPanel, "examples");
    } }
}
```

Подія ExitActionPerformed закриває тренажер якщо натиснути кнопку «Вихід». Аналогічна дія відбувається для кнопок «Завершити тренінг».

```
private void ExitActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

Якщо натиснути на кнопку «Розв'язати приклад», то програма переключиться на вкладку «Розв'язок» завдяки події StartExampleActionPerformed.

```
private void StartExampleActionPerformed(java.awt.event.ActionEvent
evt) {
    Examples.setSelectedIndex(1);
}
```

Подія `SNextStepActionPerformed` спрацьовує при натисненні кнопки «Наступний крок» і робить перевірку наданої відповіді. Якщо все правильно, то відбувається перехід до наступного кроку, якщо ні – виводиться повідомлення про помилку (Див. Додаток А).

Натиснувши «Наступний приклад» завдяки `NextExampleActionPerformed` виведеться умова наступного прикладу. При розв’язанні останнього прикладу кнопка буде мати назву «Розпочати спочатку» і відобразиться умова першого прикладу.

```
private void NextExampleActionPerformed(java.awt.event.ActionEvent
evt) {
    buttonGroup1.clearSelection();
    jCheckBox1.setSelected(false);
    jCheckBox2.setSelected(false);
    jCheckBox3.setSelected(false);
    jCheckBox4.setSelected(false);
    if(example==3) NextExample.setText("Розпочати спочатку");
    if(example==4) {
        NextExample.setText("Наступний приклад");
        example=1;
    } else {
        example++;
    }
    step=1;
    NextCondition();
    NextStart(example);
    NextSolution();
}
```

```
Examples.setEnabledAt(2, false);  
SNextStep.setEnabled(true);  
Examples.setSelectedIndex(0);  
}
```

4.3. Опис програми

На стартовій сторінці тренажера виводиться тема тренажера, вказується автор і керівник. Надається можливість перейти до теорії або практики (рис. 4.1.).

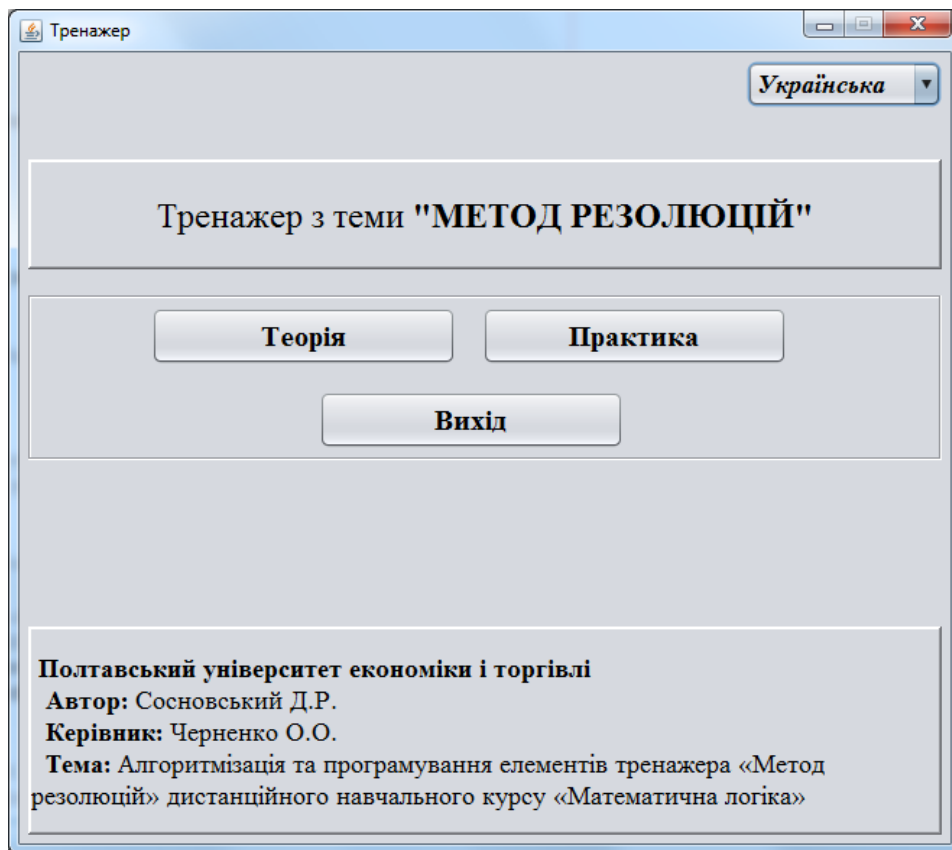


Рисунок 4.1 – Стартова сторінка тренажера

При переході до теорії відображається панель з теорією (рис. 4.2).
Кожна кнопка виводить відповідний матеріал (рис. 4.3).

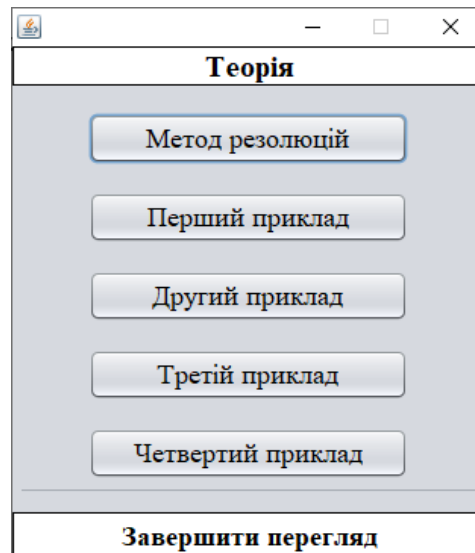


Рисунок 4.2 – Панель з теоретичним матеріалом

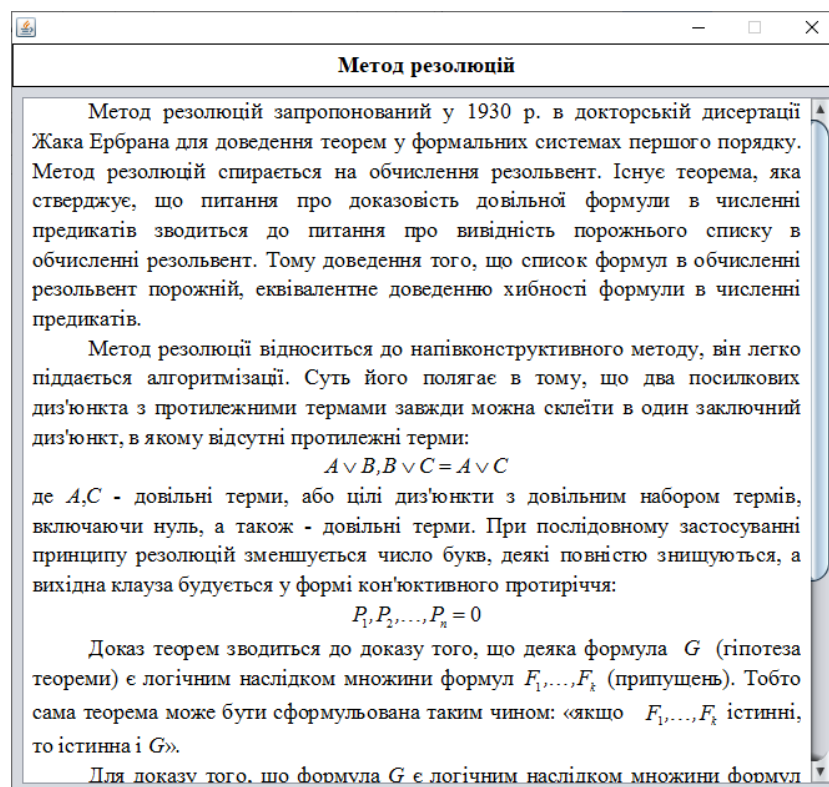


Рисунок 4.3 – Теоретичний матеріал з теми «Метод резолюцій»

Якщо було вибрано англійську версію, то стартова сторінка і вся теорія відображається на англійській мові (рис. 4.4-4.5).

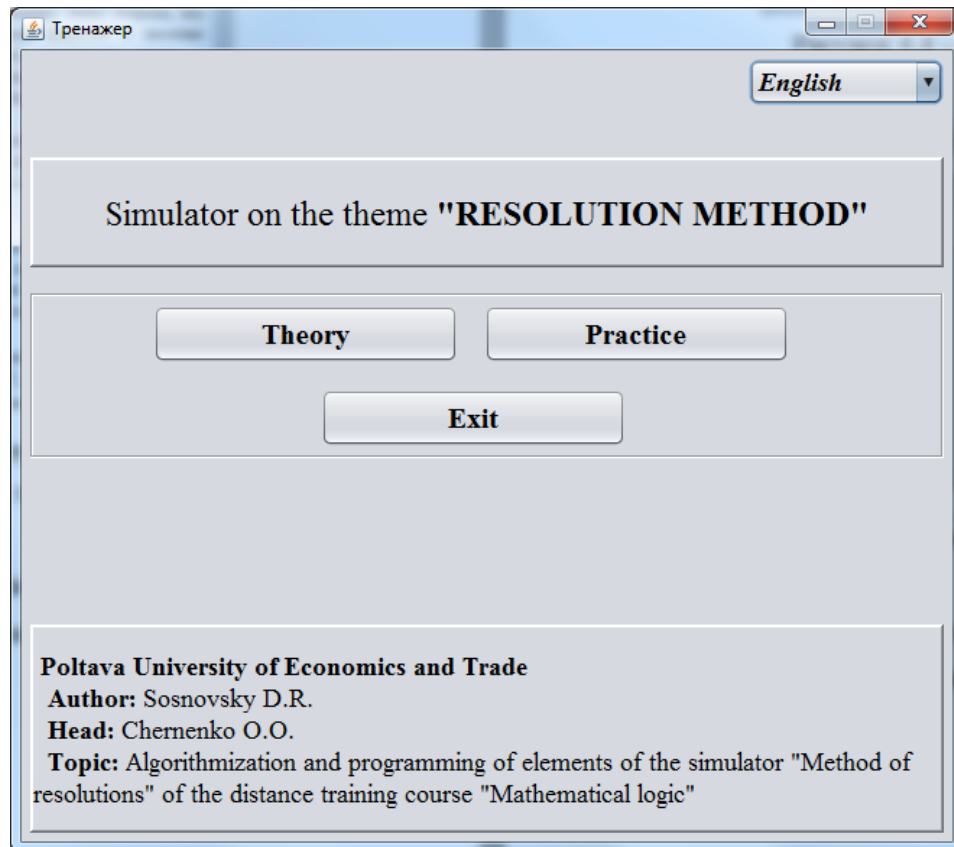


Рисунок 4.4 – Стартова сторінка на англійській мові

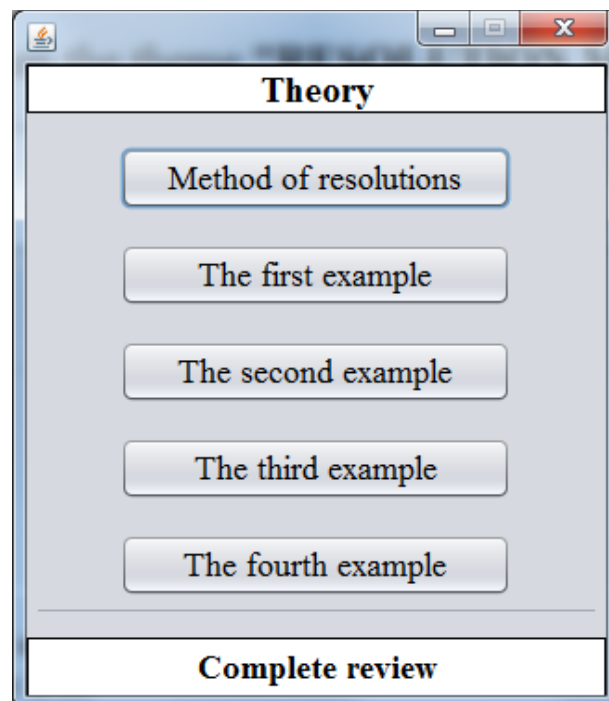


Рисунок 4.5 – Панель з теоретичним матеріалом на англійській мові

Кожний приклад починається з умови, де пропонується розв'язати його або завершити тренінг (рис. 4.6).

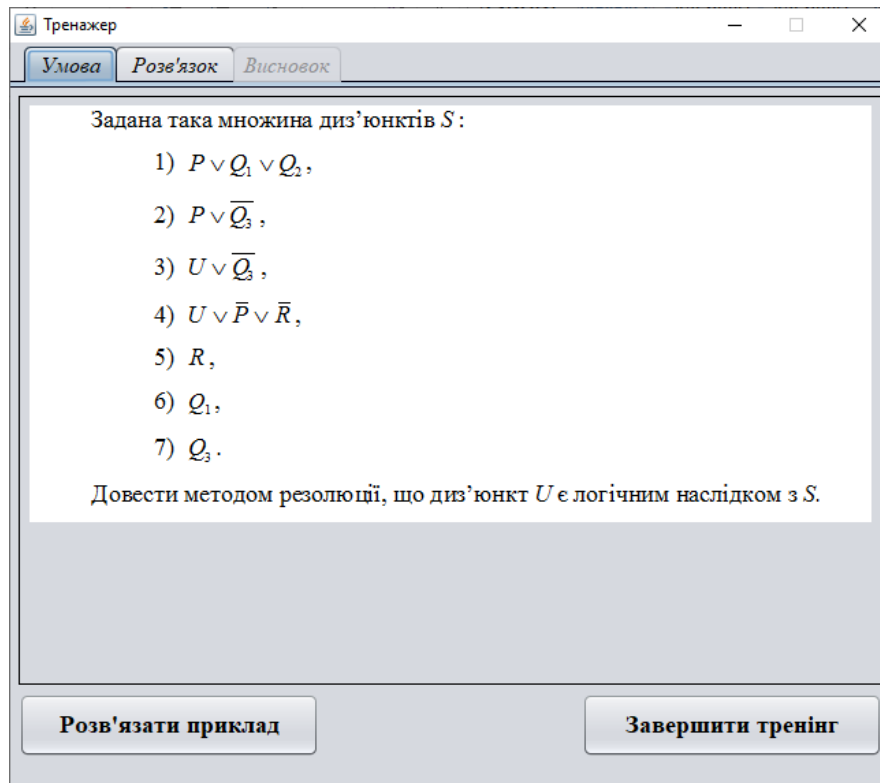


Рисунок 4.6 – Умова першого прикладу

В кожному прикладі спочатку відображається умова, а вже потім саме завдання і варіанти відповіді (рис. 4.7).

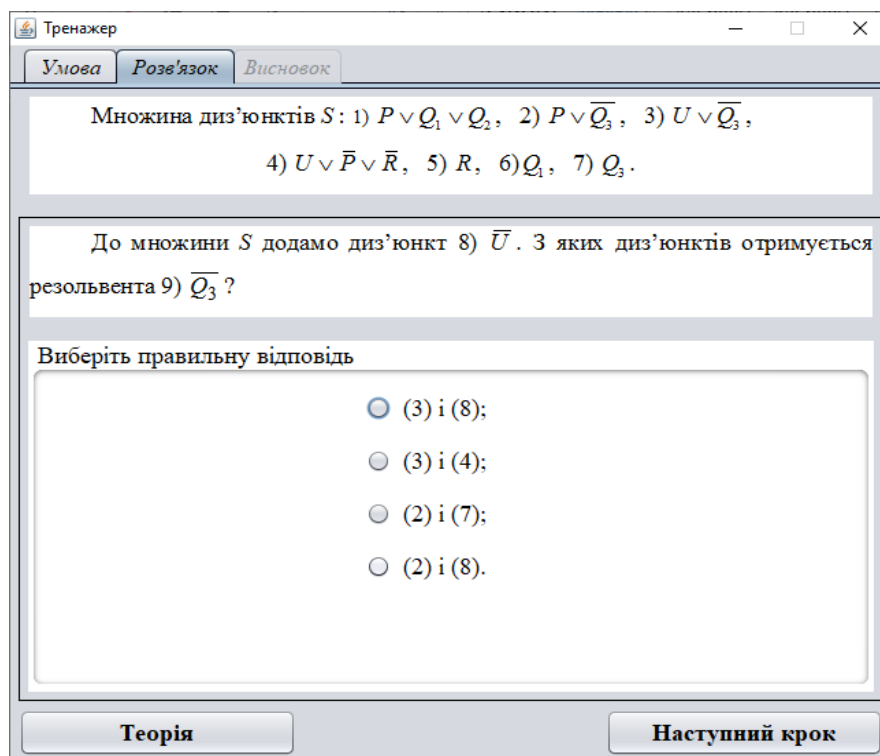


Рисунок 4.7 – Другий крок першого прикладу

Розв'язавши приклад виводиться висновок, надається можливість перейти до наступного або завершити тренінг (рис. 4.8).

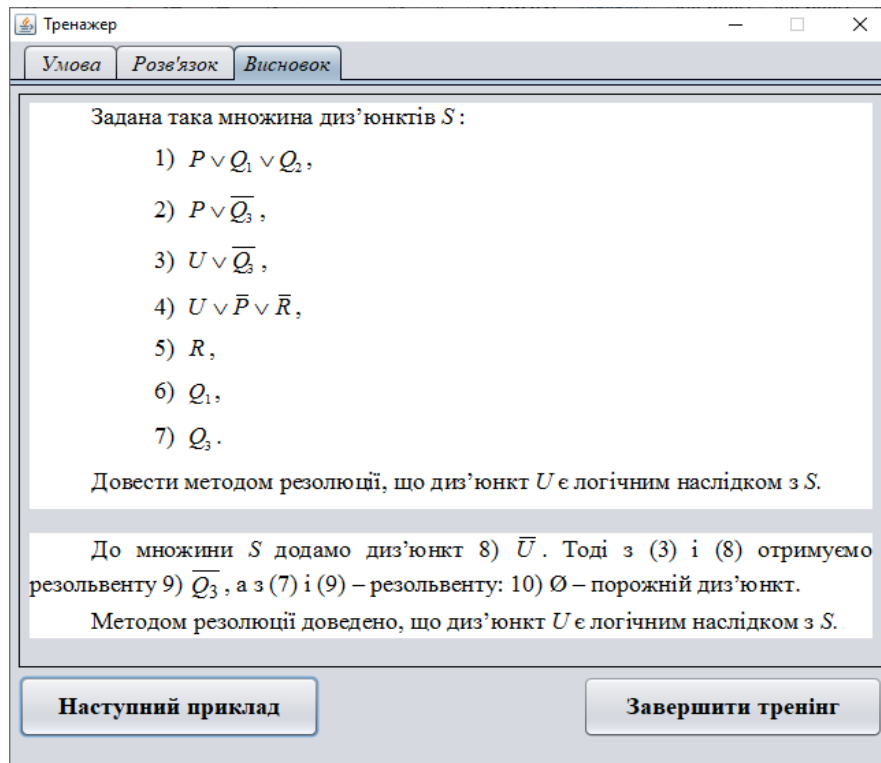


Рисунок 4.8 – Висновок першого прикладу

В третьому прикладі є завдання з вибором декількох варіантів відповідей (рис. 4.9).

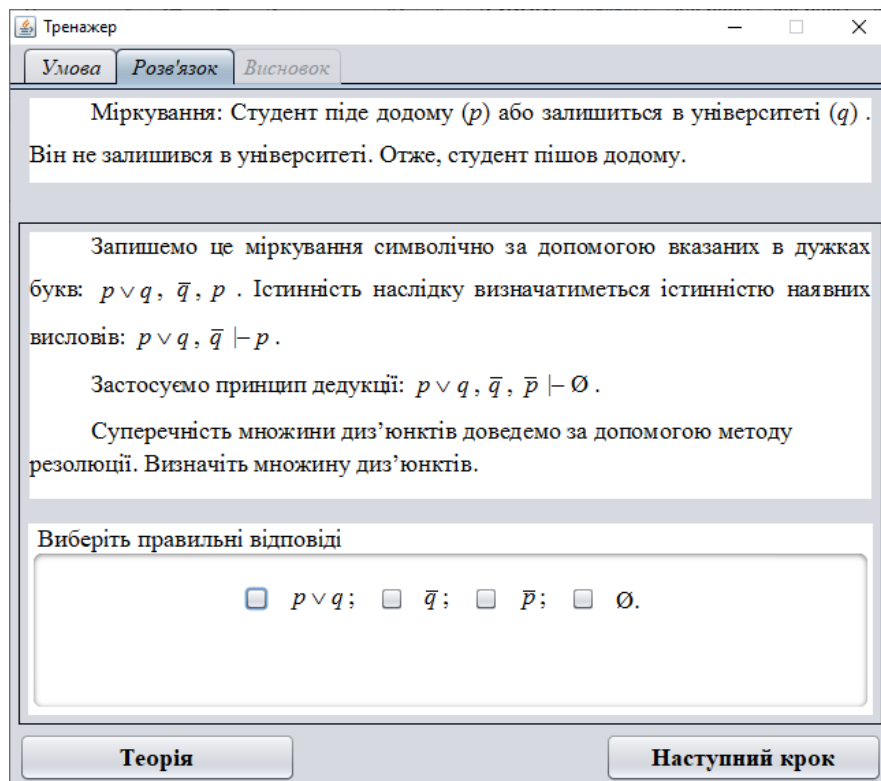


Рисунок 4.9 – Другий крок третього прикладу

При неправильній відповіді на будь-якому кроці відображається повідомлення про помилку (рис. 4.10).

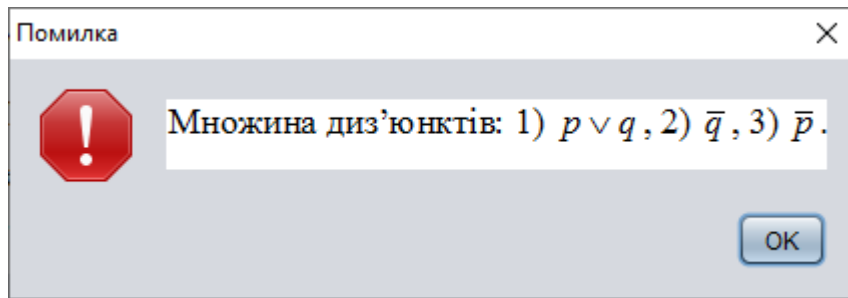


Рисунок 4.10 – Повідомлення про помилку на другому кроці третього прикладу

4.4. Інструкція по використанню тренажера

Запустивши тренажер користувачу відображається стартова сторінка з основною інформацією про нього (рис.4.1).

Щоб ознайомитися з теоретичним матеріалом слід натиснути кнопку «Теорія», після чого з'явиться панель, в якій можна переглянути потрібний матеріал чи приклад застосування методу резолюцій (рис. 4.2 - 4.3, 4.11).

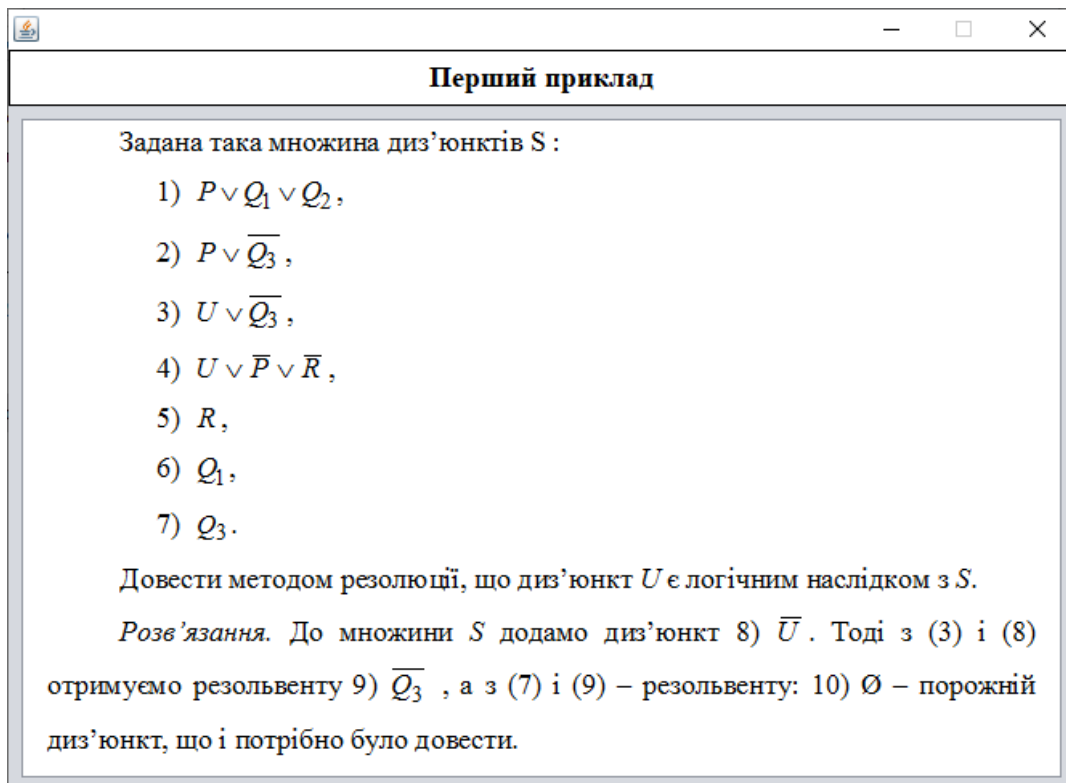


Рисунок 4.11 – Перший приклад застосування методу резолюцій

Для того щоб закрити теорію потрібно натиснути «Завершити перегляд». Кнопка «Вихід» закриває вже тренажер.

Для переходу до тренінгу необхідно натиснути кнопку «Практика», після чого користувачу виведеться умова першого прикладу (рис. 4.6). В даному випадку «Розв'язати приклад» – перейти до розв'язання, «Завершити тренінг» – закрити тренажер.

Оскільки умова прикладу відображається на першому кроці алгоритму, то на другому кроці вже виводиться завдання (рис. 4.7). Потрібно вибрати відповідь та натиснути кнопку «Наступний крок».

Якщо відповідь правильна, то відбудеться перехід до наступного кроку (рис. 4.12), якщо неправильна – виведеться повідомлення про помилку (рис. 4.13).

Тренажер

Умова Розв'язок Висновок

Множина диз'юнктів S : 1) $P \vee Q_1 \vee Q_2$, 2) $P \vee Q_3$, 3) $U \vee Q_3$,
4) $U \vee \bar{P} \vee \bar{R}$, 5) R , 6) Q_1 , 7) Q_3 .

До множини S додамо диз'юнкт 8) \bar{U} . Тоді з (3) і (8) отримуємо
резольвенту 9) \bar{Q}_3 . Чи доведено, що диз'юнкт U є логічним наслідком з S ?

Виберіть правильну відповідь

☐ Так, диз'юнкт U є логічним наслідком з S ;

☐ Ні, диз'юнкт U не є логічним наслідком з S ;

☐ Ні, потрібно продовжити доведення.

Теорія Наступний крок

Рисунок 4.12 – Третій крок першого прикладу

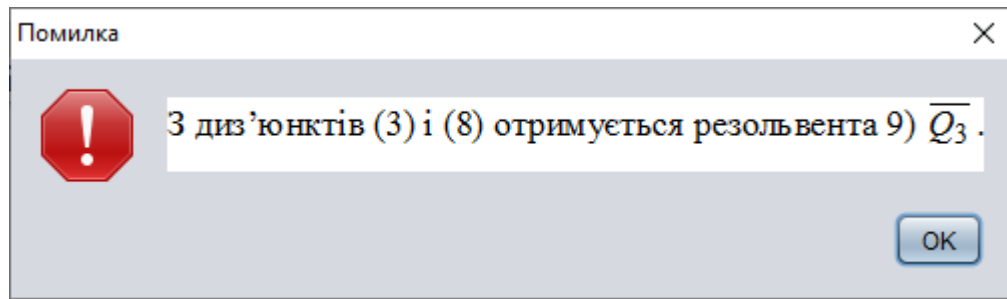


Рисунок 4.13 – Повідомлення про помилку на другому кроці першого прикладу

Таким чином слід відповісти на всі завдання. Після цього відбудеться перехід до висновку (рис. 4.8). Щоб завершити тренінг необхідно натиснути «Завершити тренінг», щоб продовжити – «Наступний приклад». При цьому виведеться умова наступного прикладу (рис. 4.14).

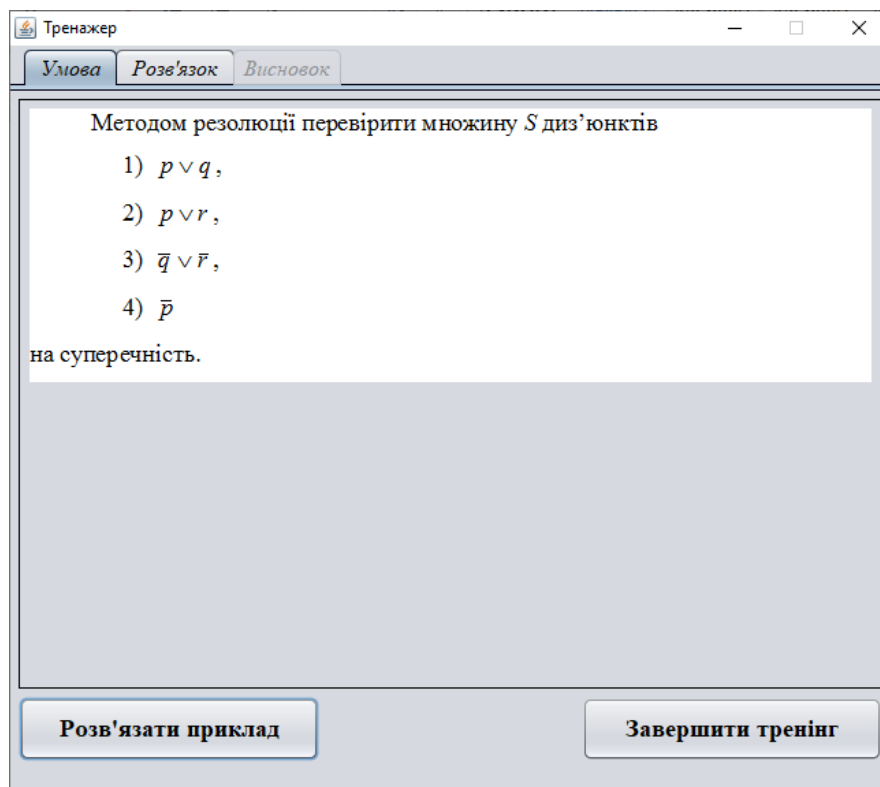


Рисунок 4.14 – Умова другого прикладу

Розв'язавши всі приклади користувачу пропонується розпочати спочатку або завершити тренінг (рис. 4.15).

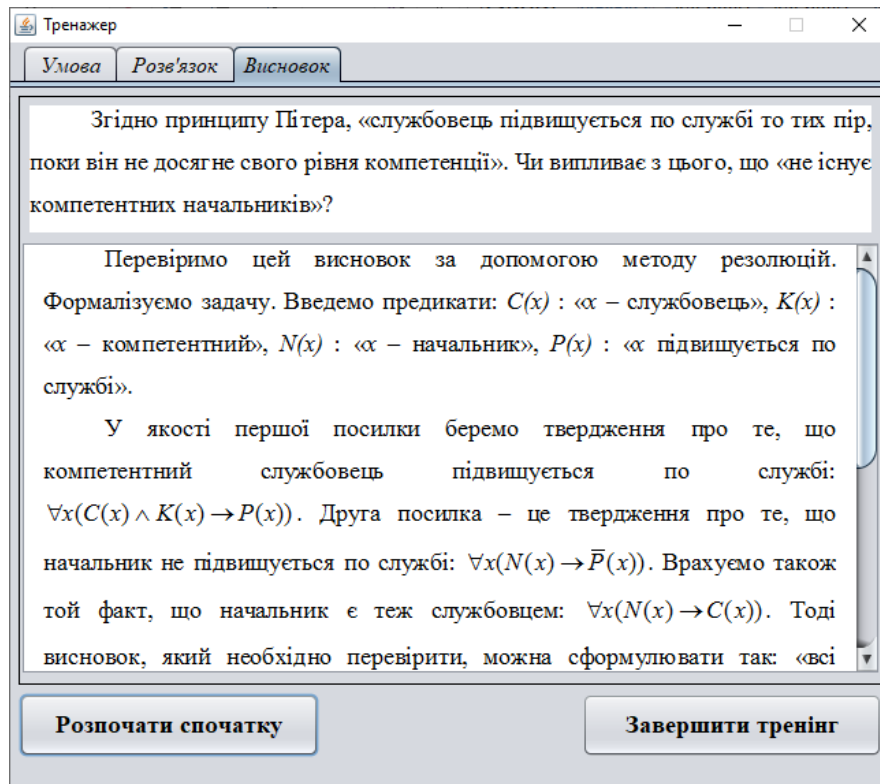


Рисунок 4.15 – Висновок четвертого прикладу

ВИСНОВКИ

Отже, під комп'ютерним тренажером розуміється програмний засіб для формування умінь і навичок з певної діяльності, а також розвитку пов'язаних із цією діяльністю здібностей. Це поняття входить до більш загальних понять програмно-педагогічного засобу та електронного засобу навчального призначення, які відрізняються як за функціональними можливостями, так і за сферою застосування.

Результатом виконання роботи є тренажер з теми «Метод резолюцій» дистанційного навчального курсу «Математична логіка».

Основні результати роботи:

- розглянуто тренажери схожої тематики з математичних дисциплін;
- описано всі аспекти оглянутих тренажерів та актуальність теми роботи;
- розглянуто теоретичні відомості про метод резолюцій;
- розглянуто специфіку застосування методу резолюцій в численні висловлень і численні предикатів;
- розроблено алгоритм тренажера, що дозволить закріпити знання з теми «Метод резолюцій».

Першим кроком було розглянуто тренажер з теми «Переставні многогранники» дистанційного курсу «Елементи комбінаторної оптимізації». Наступний – з теми «Градiєнтний метод» дистанційного курсу «Методи оптимізації та дослідження операцій». Останнім був теж тренажер з дисципліни «Методи оптимізації та дослідження операцій», що ознайомлює користувача з методом гілок та меж в задачі про найкоротший шлях.

В алгоритмі покроково описано всі дії при розв'язанні прикладів, де необхідно застосувати метод резолюцій в численні висловлень та метод

резолюцій в численні предикатів. Після цього створено блок-схему розробленого алгоритму.

До головних причини для використання мови Java і середовища IDE NetBeans відносяться:

- Відразу готова до роботи;
- Безкоштовно і з відкритим вихідним кодом;
- Потужний GUI Builder;
- Підтримка стандартів і платформ Java;
- Засоби профілювання і налагодження;
- Підтримка динамічної мови;
- Платформа, що розширюється;
- Проекти, які настроюються;
- Спеціальна підтримка.

Робота над тренажером почалась з дизайну. Спочатку розроблено основні панелі:

- StartPanel – стартова сторінка;
- Examples – приклади;
- TheoryPanel – панель з вибором матеріалу;
- TheoryMain – метод резолюцій;
- TheoryExample1 – перший приклад;
- TheoryExample2 – другий приклад;
- TheoryExample3 – третій приклад;
- TheoryExample4 – четвертий приклад.

Потім було підключено бібліотеки і оголошено змінні.

Для відображення повідомлення про помилку створено функцію ShowError(int ex, int i). Щоб вивести умову наступного прикладу реалізовано функцію NextCondition().

Функція `NextSolution()` перевіряє чи досягнуто останнього кроку алгоритму прикладу. Якщо так, то виводиться розв'язок, якщо ні – наступний крок.

Для кожної кнопки було реалізовано відповідну подію.

При виборі мови відбувається подія `LanguageItemStateChanged`. Вона змінює вміст стартової сторінки і теоретичного матеріалу залежно від вибраної мови.

Результати роботи опубліковано в матеріалах науково-практичного семінару «Комп'ютерні науки і прикладна математика (КНіПМ-2019)» та у Збірнику наукових статей магістрів Вищого навчального закладу Укоопспілки «Полтавський університет економіки і торгівлі».

Застосування нових інформаційних технологій, зокрема комп'ютерних тренажерів, у навчальному процесі дозволить об'єднати різні підходи для отримання найкращих результатів у навчанні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання дипломної роботи для студентів ступеня магістра спеціальності 122 «Комп'ютерні науки» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2018. – 35 с.
2. Дистанційний курс «Елементи комбінаторної оптимізації» // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКИ «Полтавський університет економіки і торгівлі». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=1005>
3. Марченко Д. А. Алгоритмізація тренажеру з теми "Многогранник переставлень" дистанційного навчального курсу "Елементи комбінаторної оптимізації" / Д. А. Марченко, О. О. Ємець // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/2949>
4. Марченко Д. А. Програмування навчального тренажера з теми "Переставні многогранники" дистанційного навчального курсу "Елементи комбінаторної оптимізації" / Д. А. Марченко, О. О. Ємець // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 16–18 березня 2017 р.). – Полтава: ПУЕТ, 2017. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/5421>
5. Дистанційний курс «Методи оптимізації та дослідження операцій (Частина 2) (2019) // Головний центр дистанційного навчання вищого навчального закладу УКООПСПЛКИ «Полтавський університет економіки і торгівлі». – Режим доступу: <http://www2.el.puet.edu.ua/st/course/view.php?id=1287>
6. Потерайло О. О. Алгоритмізація тренажеру з теми "Гradientний метод" дистанційного курсу "Методи оптимізації та дослідження операцій" /

О. О. Потерайло // Інформатика та системні науки (ІСН-2016): матеріали VII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 10–12 берез. 2016 р.). – Полтава: ПУЕТ, 2016. – Режим доступу:

<http://dspace.puet.edu.ua/handle/123456789/2948>

7. Потерайло О. О. Програмування навчального тренажера для градієнтного методу оптимізації нелінійних функцій дистанційного навчального курсу "Методи оптимізації та дослідження операцій" / О. О. Потерайло // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 16–18 березня 2017 р.). – Полтава: ПУЕТ, 2017. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/5483>
8. Нікольський Ю.В. Дискретна математика / Ю.В. Нікольський, В.В. Пасічник, Ю.М. Щербина. – К.: Видавнича група BVH, 2007. – 368 с.
9. Таран Т.А. Основи дискретної математики / Т.А. Таран. – К.: Просвіта, 2003. – 288 с.
10. Жук П. Ф. Математична логіка та теорія алгоритмів : практикум / уклад.: / П. Ф. Жук – К. : НАУ, 2014. – 21 с.
11. Сосновський Д.Р. Алгоритмізація та програмування елементів тренажера «Метод резолюцій» дистанційного навчального курсу «Математична логіка» / Д.Р. Сосновський // Комп'ютерні науки і прикладна математика (КНіПМ-2019): матеріали науково-практичного семінару. Випуск 3 / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – 83 с. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/7043>
12. Java технологія. Основні риси та перспективи застосування [Електронний ресурс]. – Режим доступу: <http://ua.textreferat.com/referat-8239-2.html>
13. Главные причины для перехода на среду IDE NetBeans [Електронний ресурс]. – Режим доступу: https://netbeans.org/switch/why_ru.html

14. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.